

Robustness via Cross-Domain Ensembles

Supplementary Material

1. Overview

We provide further details and evaluations in the supplementary material, as outlined below.

1. Quantitative evaluations:

- Performance on distorted and clean data (Sec. 2.1 and 2.2).
- Additional ablations with deep ensembles (Sec. 2.3).
- Importance of the paths employed and their performance against corruptions (Sec. 2.4).
- Effect of the chosen distortions for sigma training (ST) on the performance on unseen distortions (Sec. 2.5).
- Robustness of sigmas to distribution shifts (Sec. 2.6).
- Absolute errors for ImageNet-C, CIFAR-100-C and CIFAR-10-C (Sec. 2.7).

2. Qualitative results:

- **Video evaluations** performed by applying the proposed method and the baselines frame-by-frame to several videos (Sec. 3.1).
 - Additional results on distorted Replica images and external queries (Sec. 3.2).
3. Further method details: description of learning based merging (Sec. 4.1), consistency training (Sec. 4.2), adversarial attack setup (Sec. 4.3), and blind guess baseline formulation (Sec. 4.4).
 4. Definition of middle domains (Sec. 5).
 5. Visualizations of Common Corruption [2] for a sample image (Sec. 6).
 6. Our development **code** (For reference only. Without full documentation or cleanup. Attached as a separate folder to the supplementary material.)

2. Quantitative Results

2.1. Performance on Distorted Data

Performance under common corruptions and dataset shift: Figure 8 shows 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. In Table 1 we report results on Replica with common corruptions (Replica-C) and Habitat. They show supportive trends for our method, similar to Taskonomy results in Fig. 6 of main paper.

Method	Replica-C			Habitat
	Normals	Reshading	Depth	Depth
Blind guess	15.0	17.0	5.3	8.1
Baseline	6.2	12.1	3.7	5.2
Deep ensembles	6.0	11.6	3.3	4.6
Ours	5.1	10.1	2.4	4.3

Table 1: **Robustness on common corruptions and dataset shift.** The table provides quantitative evaluation on Replica-C and Habitat. ℓ_1 losses are reported, multiplied by 100 for readability. Our proposed method outperforms the baselines.

Perceptual losses: Similar to the evaluation setup by [10], besides the direct metric, we also report perceptual errors in Figure 6 for a more complete evaluation. Such metrics evaluate the same prediction in a different representation space (e.g. depth $\rightarrow$ normal) to give a non-uniform weighting to pixels depending on their properties. The proposed method yields notably lower perceptual errors for all target domains compared to baselines.

Additional ablation results: Figure 7 demonstrates the performance when the proposed method does not use sigma training and training with consistency constraints. This evaluation is performed for both target and perceptual domains. We also provide the results when deep ensembles are trained with consistency constraints. Our method significantly outperforms the baselines in all evaluations. In Figure 9 we further demonstrate the performance for each distortion.

2.2. Performance on Clean Data

In Table 4, we report quantitative evaluations on the *undistorted* Taskonomy [11] and Replica [7] datasets for the target domains depth, reshading, and surface normals. Our method variants yield similar performance on in-distribution data, while also being more robust against distribution shifts as shown in the main paper. This demonstrates *the robustness on out-of-distribution data did not come at the cost of degraded performance for in-distribution data*. We do not expect robust methods (ours and deep ensembles) to yield a better performance on in-distribution data as well, since they are not designed to do so and do not have a reason for that. However, we observe that they occasionally fit better to even the in-distribution data, as shown in Table 4.

2.3. Additional ablations with deep ensembles

The key role of middle domains can be further seen from Figure 2. It shows the error against distribution shift when deep ensembles is trained with consistency constraints (*cons*) and sigma training (*ST*) for Reshade task. Merging is done with both the proposed uncertainty based weighting (*inv. var. merging*) and the average of the predictions (*uniform merging*). Thus, the only remaining difference between deep ensembles and ours is in utilizing middle domains. In both cases, our method clearly outperforms.

2.4. Path Importance

Figure 10 shows the performance for individual distortions as a function of number of paths. The proposed method consistently outperforms deep ensembles and the performance improves with more paths. In Table 3 and Figure 5, we further investigate the importance of each employed path and best performing paths for the depth, reshading, and normals prediction.

2.5. Effect of Employed Distortions for Sigma Training

Here we investigate the effect of using different distortions for the sigma training (*ST*) step. For this we consider four paths: *rgb*, *greyscale*, *sharpened*, *2D edges*. We combine the predictions from these paths using the inverse variance merging with three different distortion sets for the *ST* stage: *ST #1* uses Gaussian blur and Gaussian noise similar to the results in the main paper while *ST #2* uses speckle noise and pixelate distortions and *ST #3* uses pixelate and Gaussian blur distortions. Figure 1 shows the average ℓ_1 error evaluated on nine *unseen* distortions for these variants. As can be seen, all variants perform similarly, suggesting that the *ST* step does not require handpicking the training distortions, and can be performed successfully with *any* distortion with sufficiently high intensity.

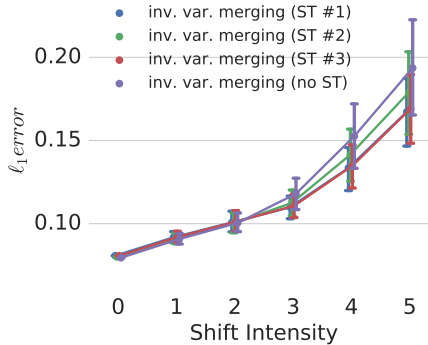


Figure 1: **Sigma training does not require handpicking the training distortions.** We consider inverse variance merging with four paths (direct, grey, sharpened, edge) for reshading prediction. *ST #1* is trained with Gaussian blur and Gaussian noise, while *ST #2* is trained with speckle noise and pixelate distortions, and *ST #3* is trained with pixelate and Gaussian blur distortions (no *ST* shows the results when there is no sigma training). The evaluations are performed over nine *unseen* distortions (shot noise, impulse noise, defocus blur, glass blur, contrast, brightness, saturate, jpeg compression, spatter) for these variants. As can be seen, sigma training performs similarly, suggesting that any distortion with sufficiently high intensity works.

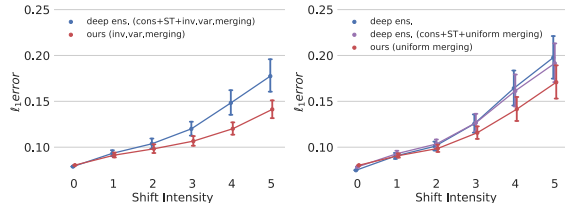


Figure 2: **Deep ensembles trained with consistency constraints and sigma training (*ST*).** The only remaining difference between deep ensembles and ours is in utilizing middle domains. Our method outperforms for both *inv. var. merging* (left) and *uniform merging* (right).

2.6. Robustness of Sigmas to Distribution Shifts

We have showed that our method returns predictions that are robust under a wide range of distribution shifts. Are our predicted uncertainties also able to generalize under distribution shifts, i.e. do we get high uncertainties when predictions get worse? To investigate this, we consider epistemic uncertainty which is used to capture the model’s uncertainty and is an indicator of distribution shifts [3]. The left plot of Figure 3 shows a scatter plot of average epistemic uncertainties against error and the right shows the average error for all epistemic sigma values less than a threshold τ . The predicted uncertainties from deep ensembles initially increases with error but does not increase past 0.15 sigma despite an increase in error, thus is overconfident, while our method shows an increasing trend.

	Method	Clean error	Avg.	Noise			Blur				Weather				Digital			
				Gauss.	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG
ImageNet	Baseline ResNet-50	24.4	76.2	73.0	74.7	78.8	79.9	92.1	81.5	82.5	75.2	75.6	64.0	59.2	65.3	90.6	74.8	75.9
	Deep ensembles	21.5	70.4	67.4	69.7	72.5	73.4	87.4	76.1	76.9	70.3	70.3	60.1	52.4	61.7	83.8	66.4	68.3
	Ours	21.6	67.9	66.6	68.6	71.2	71.7	82.1	75.6	77.3	69.1	67.2	59.1	51.3	55.8	82.1	54.4	65.9
CIFAR100	Baseline ResNet-18	24.9	48.6	76.7	68.9	72.4	40.6	50.0	45.8	44.3	41.8	47.2	40.1	30	51.0	37.3	47.1	46.1
	Deep ensembles	21.0	44.1	74.1	65.2	70.1	35.9	45.2	40.8	39.4	37.1	42.3	35.7	25.5	47.0	32.4	42.2	40.4
	Ours	20.3	41.8	69.8	60.5	59.5	35.8	45.2	41.2	40.0	36.9	40.7	37.1	23.9	46.6	32.7	39.0	36.4
CIFAR10	Baseline ResNet-26	7.9	26.3	55.5	44.8	20.5	19.9	26.6	26.7	24.7	18.4	21.5	18.2	10.3	30.8	18.3	32.8	20.2
	Deep ensembles	6.5	23.3	46.5	36.1	33.9	18.4	22.7	23.7	22.3	16.3	18.0	15.0	7.9	28.0	15.1	29.1	16.8
	Ours	6.7	20.7	40.5	31.6	27.9	15.4	19.9	21.5	18.4	15.8	16.3	16.0	8.0	25.8	13.8	26.2	13.9

Table 2: **Robustness on ImageNet-C, CIFAR-100-C, and CIFAR-10-C.** Error on clean and distorted data. For all metrics, lower is better. All methods are trained only on clean ImageNet, CIFAR-100, and CIFAR-10 training data, respectively. Our method performs noticeably better compared to deep ensembles and a single model baseline ResNet.

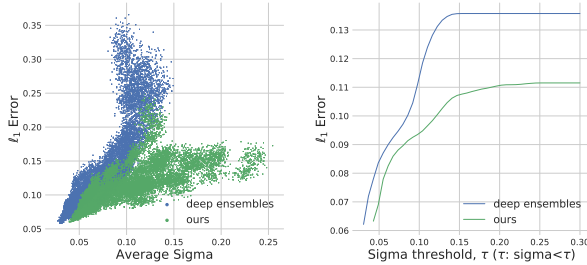


Figure 3: **Generalization of predicted uncertainties.** We compute the average ℓ_1 error and epistemic uncertainties for 11 unseen distortions and 5 levels of intensities for Reshade. Each point is an average over 64 images. Our method is able to return high uncertainty when predictions are poor.

2.7. Performance on Imagenet-C, CIFAR-100-C, and CIFAR-10-C

The absolute errors on ImageNet-C, CIFAR-100-C, and CIFAR-10-C are given in Table 2. For deep ensembles and our method, we use the same backbone as the baseline ResNet model. Our method significantly outperforms the baselines for all three classification tasks.

3. Qualitative Results

3.1. Video Evaluation

We perform video evaluations on distorted and undistorted video clips to further evaluate the performance of our method against several baselines. We strongly recommend watching them.

3.2. Distorted Images and External Queries

We provide qualitative results in Figure 11 on out of distribution data that is markedly different from those seen during training to demonstrate the generalizability of our method. Furthermore, in Figures 12,13,14,15 we provide additional prediction results for distorted Replica test images for normals, reshading, and depth prediction.

4. Further Method Details

4.1. Formulation for Learning Based Merging

Multi-modal predictions: For ill-posed problems such as ours, adopting mixture models allows us to capture inherent ambiguity in the data by assuming that there are several possible distributions that could have generated the observed data (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 in the main paper, we estimate the parameters of these K distributions, $\{\hat{z}_i, \hat{s}_i\}_{i=1}^K$, then function m learns to output the mixing weights given these set of parameters. Our final distribution is a mixture of Laplacians,

$$h_{Mix}(z) = \sum_i \hat{w}_i h(z|\hat{z}_i, \hat{s}_i) \quad (1)$$

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 $\mathcal{L}_m = \frac{1}{N} \sum_{n=1}^N -\log h_{Mix}(z_n)$.

Mixture-weights: The final prediction is computed as the weighted average of mixture components’ means using the weights $\hat{w}_i$ as in Eq. 1. We approximate the weights for a path to be proportional to its mixture probability distribution function (p.d.f.) evaluated at its estimate $\hat{z}_i$, i.e. $\hat{w}_i \propto h_{Mix}(\hat{z}_i)$. To enforce that h_{Mix} is a p.d.f, we require $\hat{w}_i \geq 0$ and $\sum_i \hat{w}_i = 1$.

Learning based merging: We consider a stacking model [9] that obtains the final predictions given the outputs from each path using the predicted mixture weights $\hat{w}_i$. It has the advantage that the loss is over the entire image, thus, taking into account its spatial structure. This option is denoted as *Network merging*.

4.2. Training with Cross-Task Consistency Loss

The proposed ensembling method can be further augmented with “cross-task consistency constraints” [10], to ensure that predictions from the different paths are in cross-task agreement. While this is not a fundamental step for the

Domain \ Best # paths	1	2	3	4	5	6	7	8
Direct	×	D,R	D,R,N	D,R,N	D,R,N	D,R,N	D,R,N	D,R,N
2D Edge	×	×	D,N	D,R,N	D,R,N	D,R,N	D,R,N	D,R,N
Lap. Edge	×	×	×	×	×	D,N	D,N	D,R,N
Emboss	×	N	×	N	D,N	R,N	D,R,N	D,R,N
Greyscale	×	×	R	D	D,R	D,R	D,R,N	D,R,N
Low pass	D,R,N	D,R,N	D,R,N	D,R,N	D,R,N	D,R,N	D,R,N	D,R,N
Sharpened	×	×	×	R	R	R	R	D,R,N
Wavelet	×	×	×	×	N	D,N	D,R,N	D,R,N

Table 3: **Best performing paths.** This shows the best combination of paths for Depth (D), Reshade (R), and Normals (N) targets as we increase the number of available paths from 1 to 8.

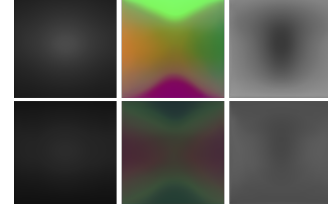


Figure 4: **Statistically informed guesses (“Blind Guess”)** on the Taskonomy dataset for depth, normal, and reshading tasks. Top row shows the predicted mean and the bottom row shows the predicted standard deviation. The predictions minimize the NLL loss on the training dataset.

Method	Taskonomy Dataset									Replica Dataset								
	Depth			Reshade			Normal			Depth			Reshade			Normal		
	Perceptual error		Direct error	Perceptual error		Direct error	Perceptual error		Direct error	Perceptual error		Direct error	Perceptual error		Direct error	Perceptual error		Direct error
	Reshade	Normal		Depth	Normal		Reshade	Depth		Reshade	Depth		Reshade	Depth		Reshade	Depth	
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
Baseline UNet (cons.)	8.121	7.409	2.198	2.360	5.494	7.252	8.121	1.784	4.957	9.172	6.192	2.090	2.535	5.288	9.051	8.915	1.377	4.405
Multi-domain	10.732	10.548	2.435	2.911	6.483	6.963	18.978	4.633	4.700	12.461	9.606	2.215	2.270	5.954	8.946	17.664	3.489	4.458
Multi-task	11.080	10.542	2.333	3.045	6.742	6.958	22.117	5.194	5.199	11.840	9.270	1.814	1.984	5.778	9.423	22.487	4.516	5.300
Deep ensembles	8.934	8.192	2.207	2.779	5.983	6.470	19.914	4.810	4.448	9.755	6.965	1.925	2.018	5.497	8.513	19.557	3.788	4.305
Deep ensembles (cons.)	8.348	7.338	2.289	2.279	5.184	6.730	8.110	1.960	4.644	9.068	5.871	2.014	2.176	4.966	8.614	9.917	1.874	4.271
Ours (Uniform merging)	8.522	7.510	2.359	2.433	5.435	6.929	8.523	2.338	4.770	8.884	5.960	1.989	2.080	5.029	8.484	10.258	2.264	4.392
Ours (Inv. var. merging)	8.542	7.877	2.301	2.339	5.389	6.787	8.342	2.044	4.685	9.443	6.798	1.980	2.194	5.051	8.451	9.539	1.734	4.220
Ours (Network merging)	8.791	8.123	2.337	2.404	5.456	6.918	8.290	1.946	4.678	9.886	6.986	2.167	2.127	5.036	8.469	9.279	1.529	4.194

Table 4: **Robustness on out-of-distribution data did not come at the cost of degrading performance on in-distribution data.** The table provides quantitative evaluation on *undistorted* Taskonomy and Replica test sets. Results are reported for depth, reshading, and normal using direct and perceptual error metrics. The perceptual metrics evaluate the target prediction in another domain, similar to [10]. 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*.

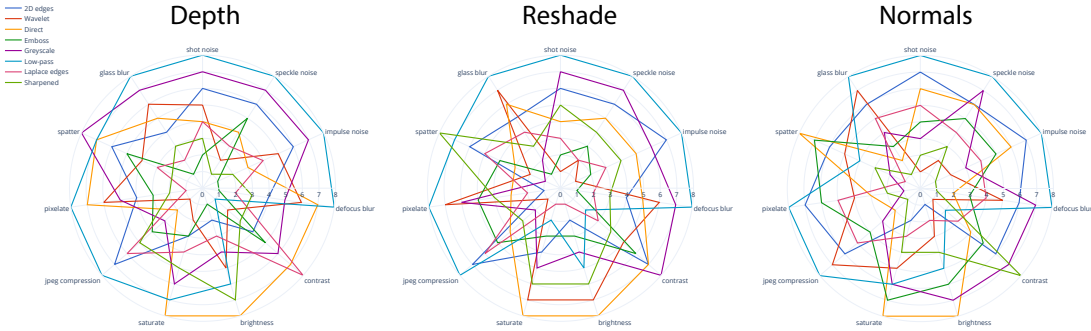


Figure 5: **Importance of each middle domain for different distortion.** The order of the best performing paths for depth, reshade, and surface normals predictions for each of the distortions, with 8 denoting the most important and 1 the least important one, is shown. This is an extension of Fig. 7b in the main paper with reshade and depth target domains.

method, it yields better accuracy especially in fine-grained regions. Following [10], we consider a set of *perceptual* loss networks on the outputs of g . 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.

4.3. Adversarial Attack/Defence Setup

To generate the I-FGSM attack [4], we apply the following:

$$X_0 = X, \quad (2)$$

$$X_{n+1} = \text{Clip}_{X,\epsilon}\{X_n + \alpha \text{sign}(\nabla J(X_n, y))\} \quad (3)$$

Similar to [4], we set $\alpha = 1$ in our experiments and the number of iterations given by $N = \min(4 + \epsilon, 1.25\epsilon)$. For

Method	Normal				Reshade				Depth			
	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
Uniform merging	7.77	9.48	11.00	13.21	15.50	18.61	21.32	27.20	4.85	5.45	5.97	7.36
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 5: **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 [4] based attacks *without adversarial training*, compared to the baselines. The last row shows the error for a model that has undergone adversarial training [5] with the *same attacks as those evaluated at test time*, hence it gives a lower bound on the error.

adversarial training, we finetune the Baseline UNets with an equal number of adversarial and clean examples in the training data. The former were generated from a I-FGSM attack with $\epsilon = (0 - 16]$.

Table 5 shows that the proposed method outperforms the baselines against corruptions generated with I-FGSM.

4.4. Blind Guess Baseline

This baseline is computed using the following formula:

$$g^* = \min_g \mathcal{L}_{NLL} \quad (4)$$

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 4.

5. Middle domain definitions

Here we define the middle domains. They can be extracted with standard libraries such as OpenCV [1], Scikit-Image [8], or implemented in Pytorch [6]. Our Pytorch implementation for them can be found [here](#).

2D edges are from the output of a Canny edge detector without non-maximum suppression.

Greyscale images are computed by taking the average over the R, G, and B channels of the input image.

Embossed: We perform the filtering using four emboss kernels with different directions and concatenate the embossed outputs.

Laplace edges: Similar to 2D edges, we perform the filtering using a Laplacian filter.

Low-pass filtered: We perform a low pass filtering with a Gaussian filter.

Sharpened: We sharpen the image by subtracting its low-pass filtered version and then adding the original image.

Wavelet: We use the Daubechies-1 with 3 levels of decomposition to obtain wavelet coefficients. We then per-

form bilinear upsampling on the low resolution coefficients to have the same size as the original image.

6. Visualizations of Common Corruption on Taskonomy data

Common Corruptions [2] used during evaluation are shown for a sample Taskonomy image in Figure 16.

References

- [1] G. Bradski. The OpenCV Library. *Dr. Dobb’s Journal of Software Tools*, 2000. 5
- [2] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 1, 5
- [3] 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
- [4] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016. 4, 5
- [5] 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. 5
- [6] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019. 5
- [7] 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, Xiqing 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. 2
- [8] Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014. 5
- [9] David H Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992. 3
- [10] 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. 1, 3, 4
- [11] 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. 2

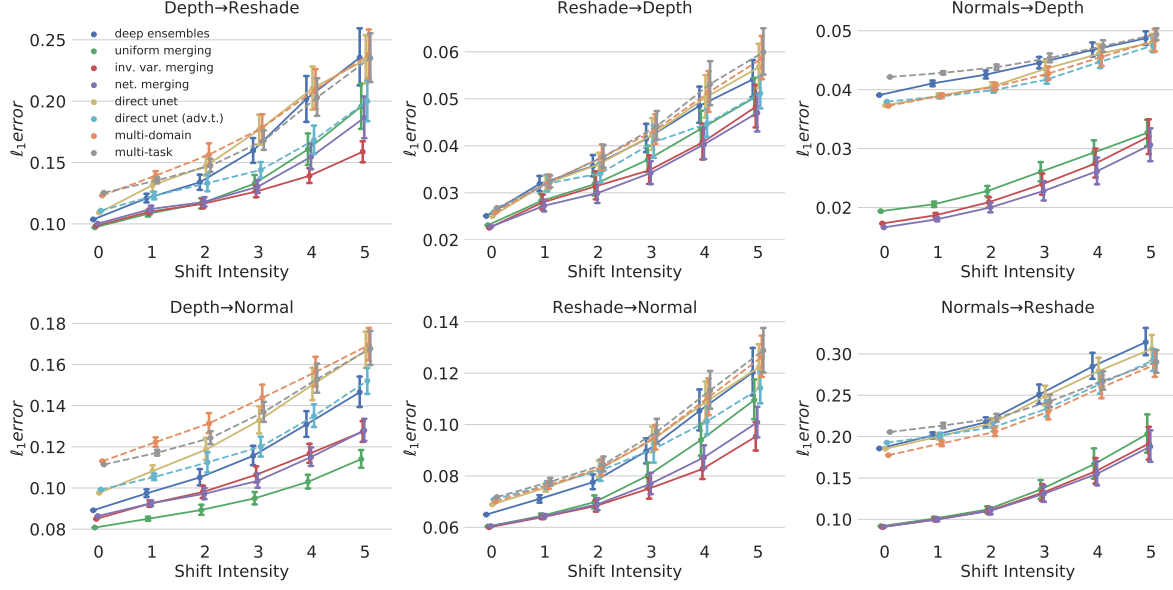


Figure 6: **Average ℓ_1 loss over 11 unseen distortions in perceptual domains.** Similar to Fig. 6 in the main paper, this shows the performance for unseen distortions where ℓ_1 losses are computed in *perceptual* domains instead, e.g. for the top left plot, loss in the reshade inferred out of the predicted depth is computed. The proposed ensembling approaches are more robust against shift with increasing intensities in perceptual domains as well.

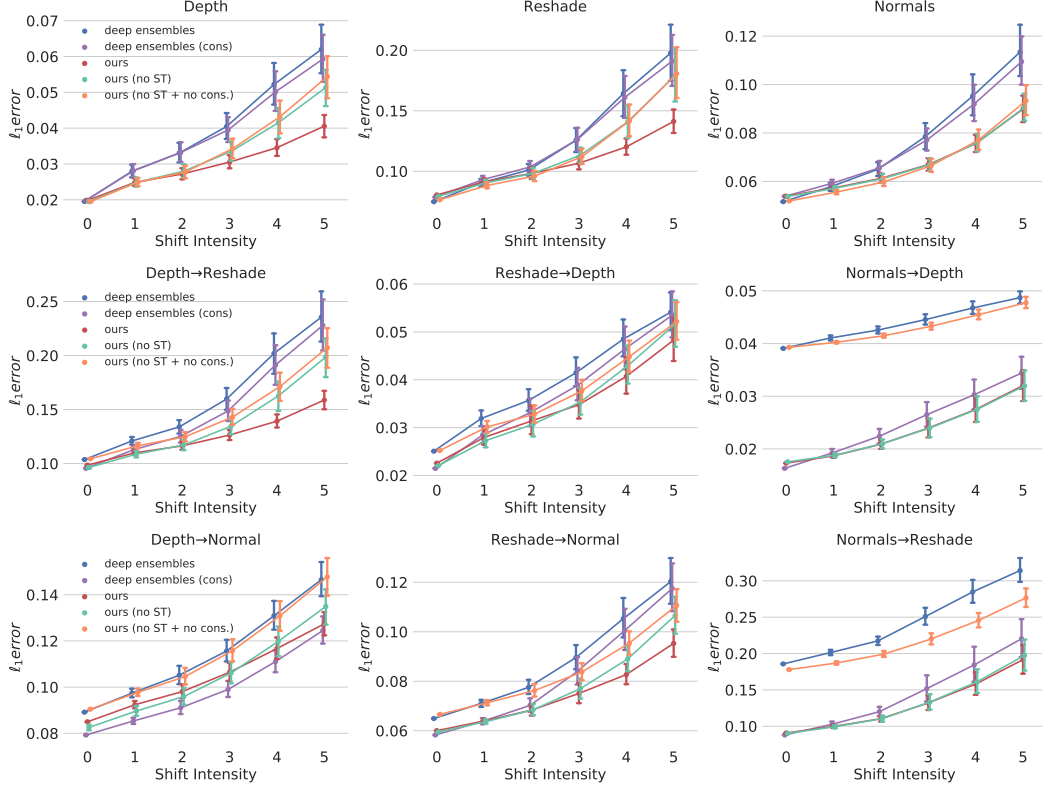
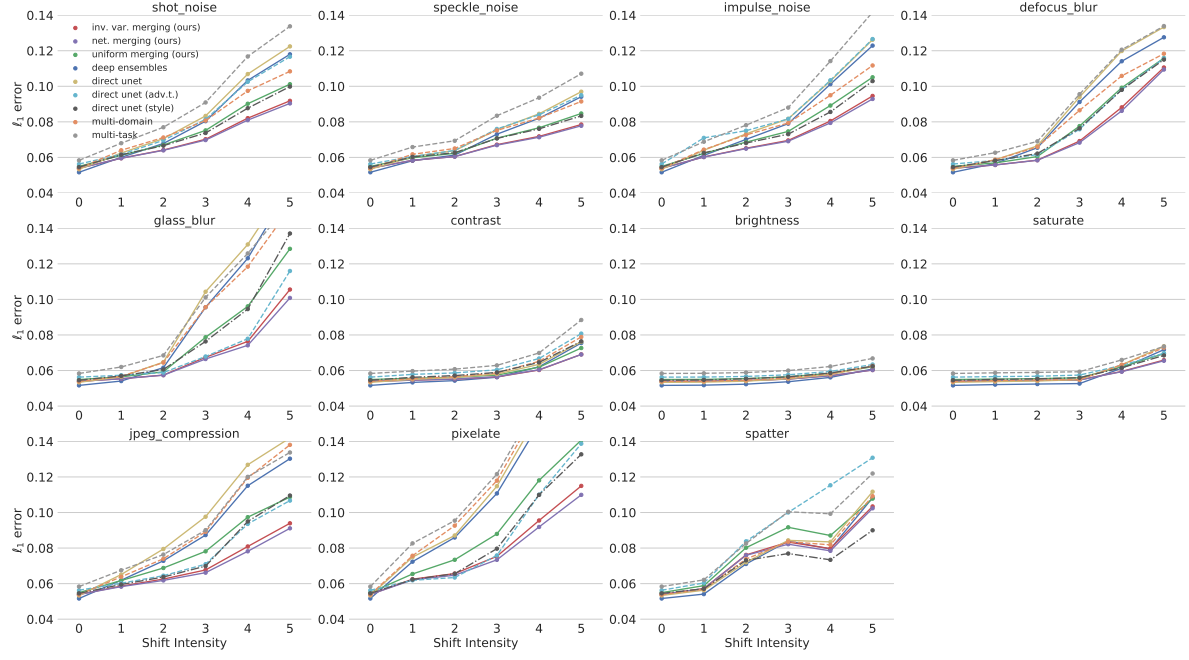
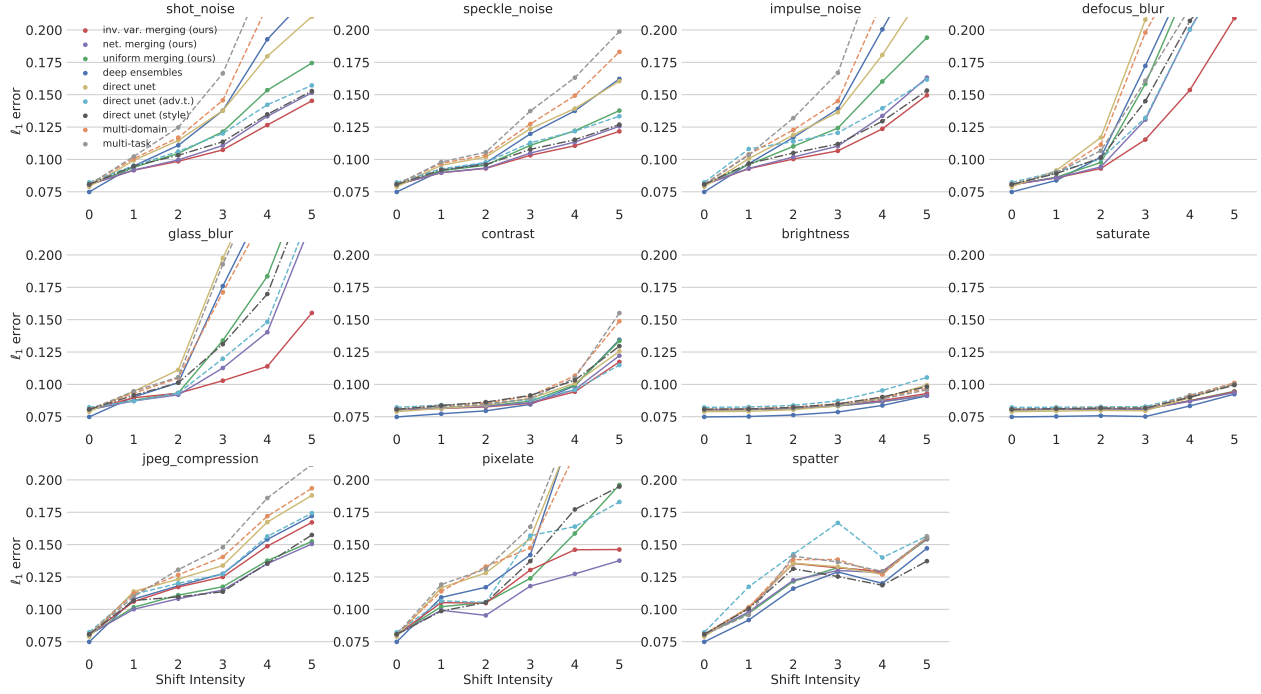


Figure 7: **Average ℓ_1 loss over 11 unseen distortions in target and perceptual domains.** This shows the performance in target and perceptual domains when the proposed method does not use sigma training or training with consistency constraints.

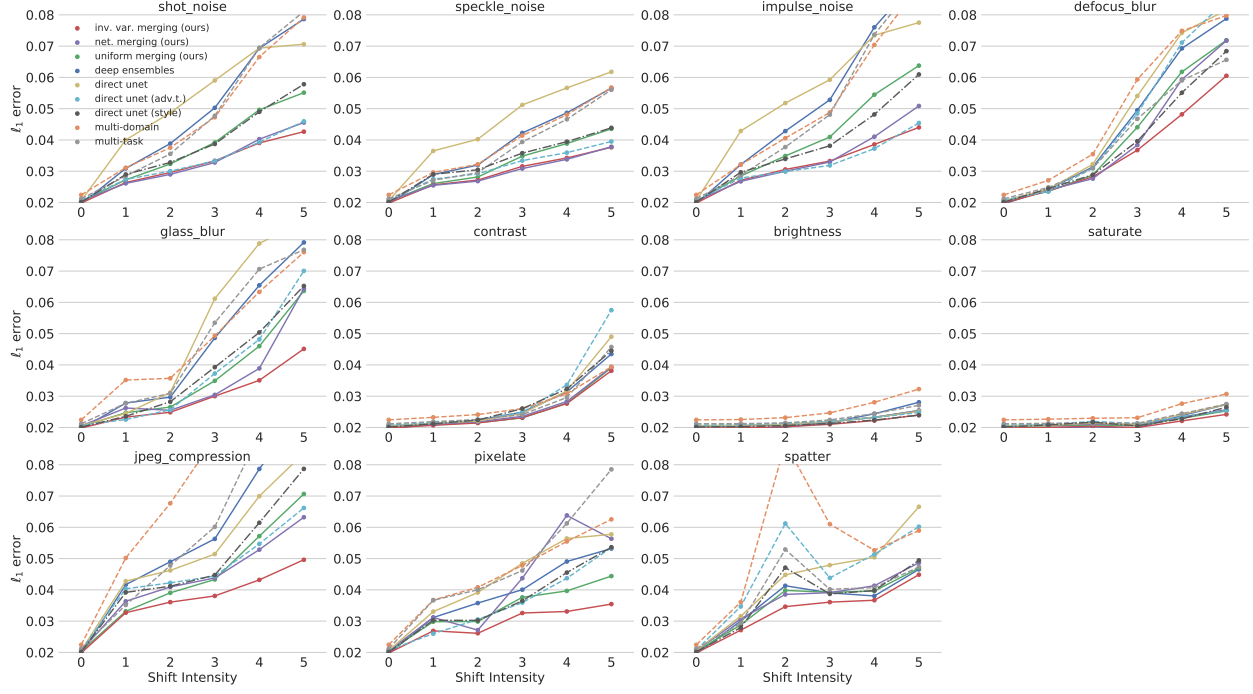


(a) Normal



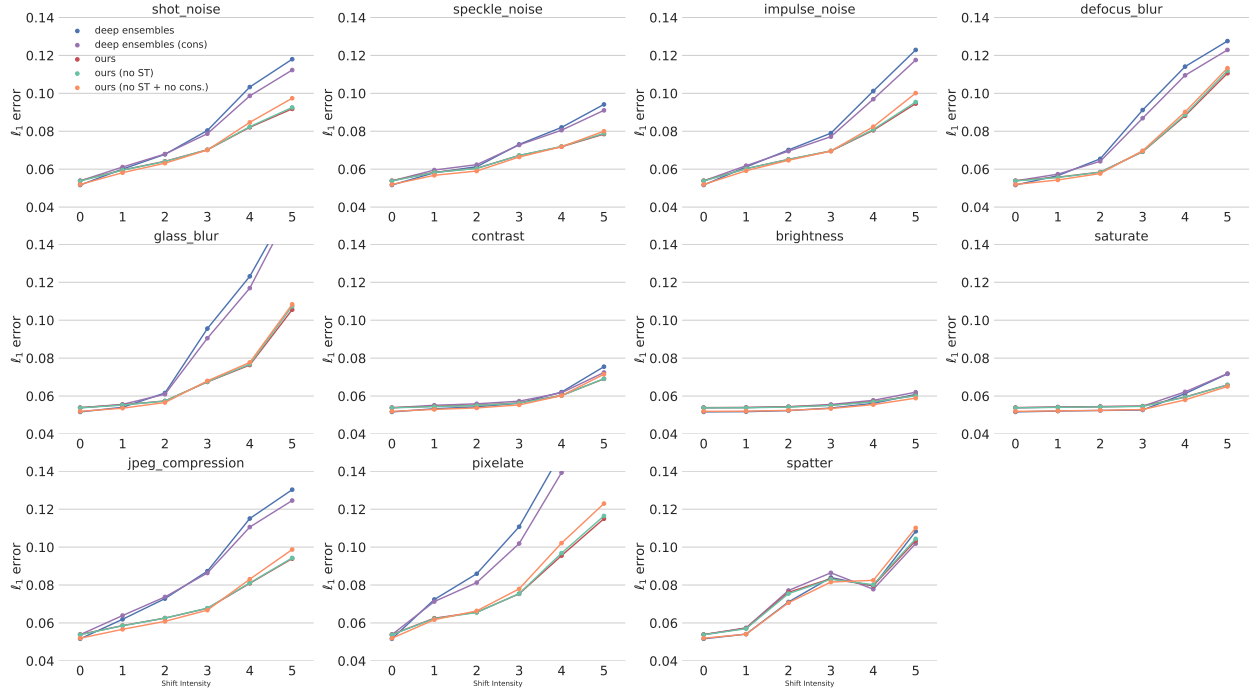
(b) Reshade

Figure 8: ℓ_1 error for each distortion for our method against several baselines. This figure is an extension of Fig. 6 in the main paper for individual distortions.



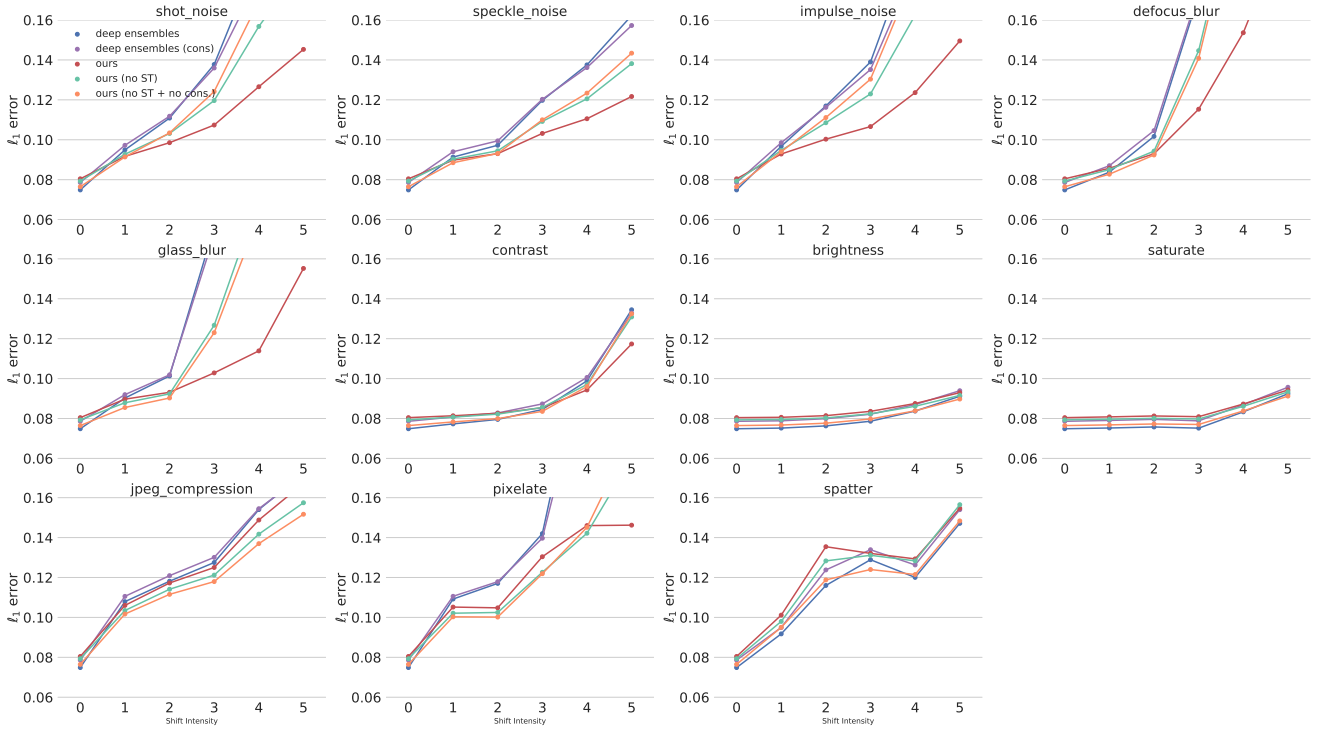
(c) Depth

Figure 8: ℓ_1 error for each distortion for our method against several baselines. This figure is an extension of Fig. 6 in the main paper for individual distortions.

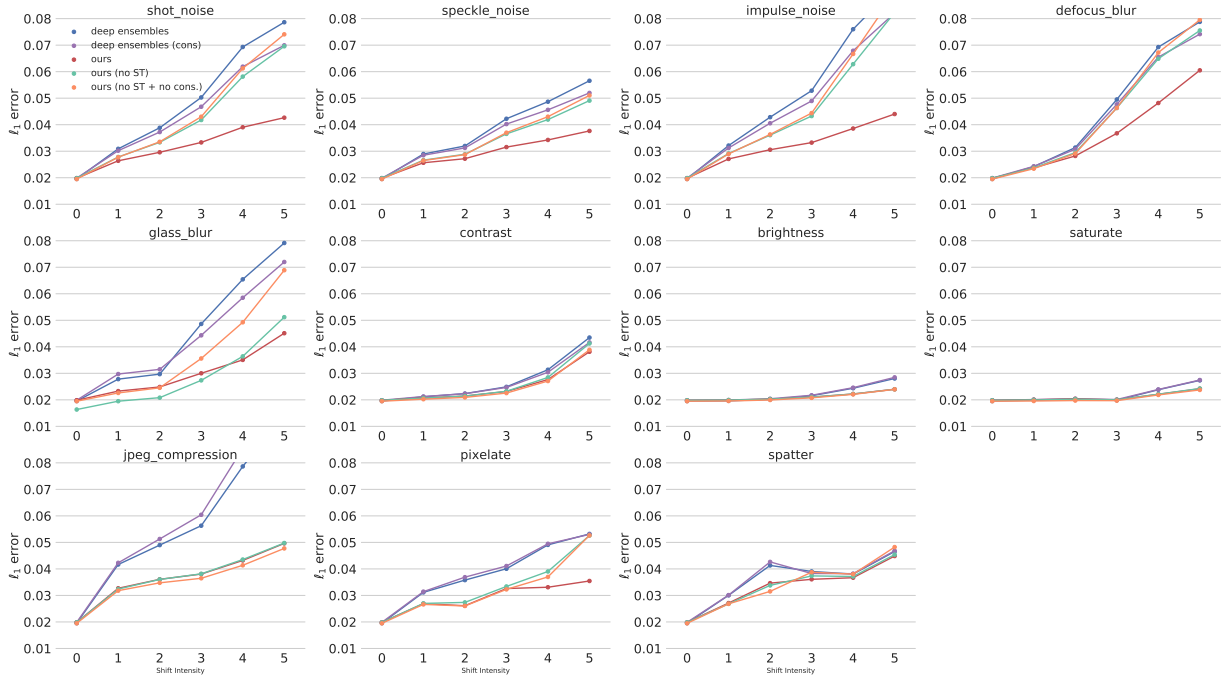


(a) Normal

Figure 9: ℓ_1 error for each distortion for our method without sigma training and consistency training stages. This figure is an extension of Fig. 7a (top) in the main paper for individual distortions.

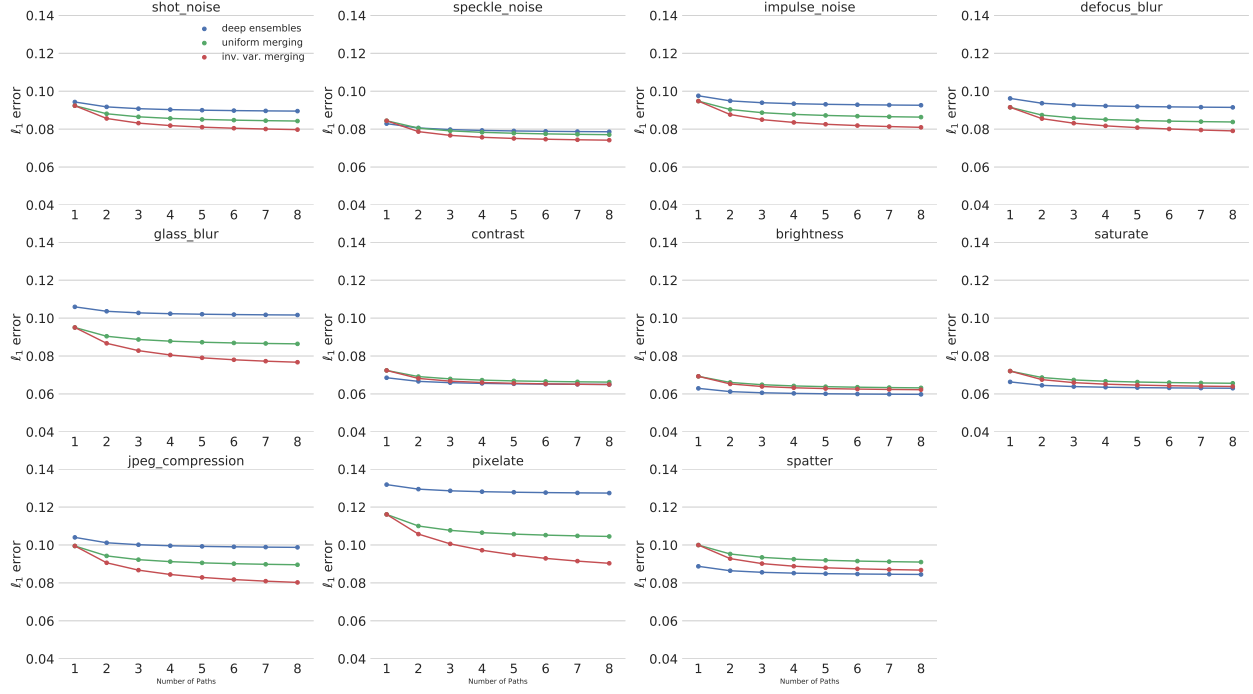


(b) Reshade

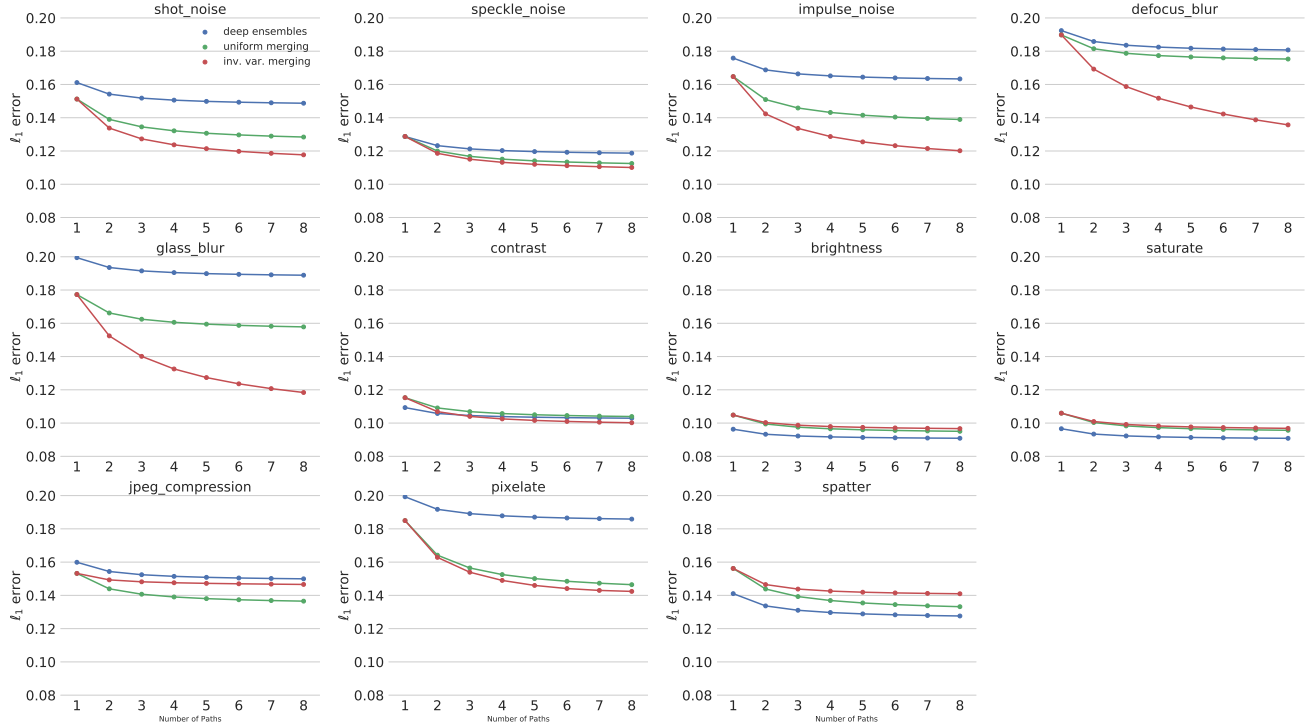


(c) Depth

Figure 9: ℓ_1 error for each distortion for our method without sigma training and consistency training stages. This figure is an extension of Fig. 7a (top) in the main paper for individual distortions.

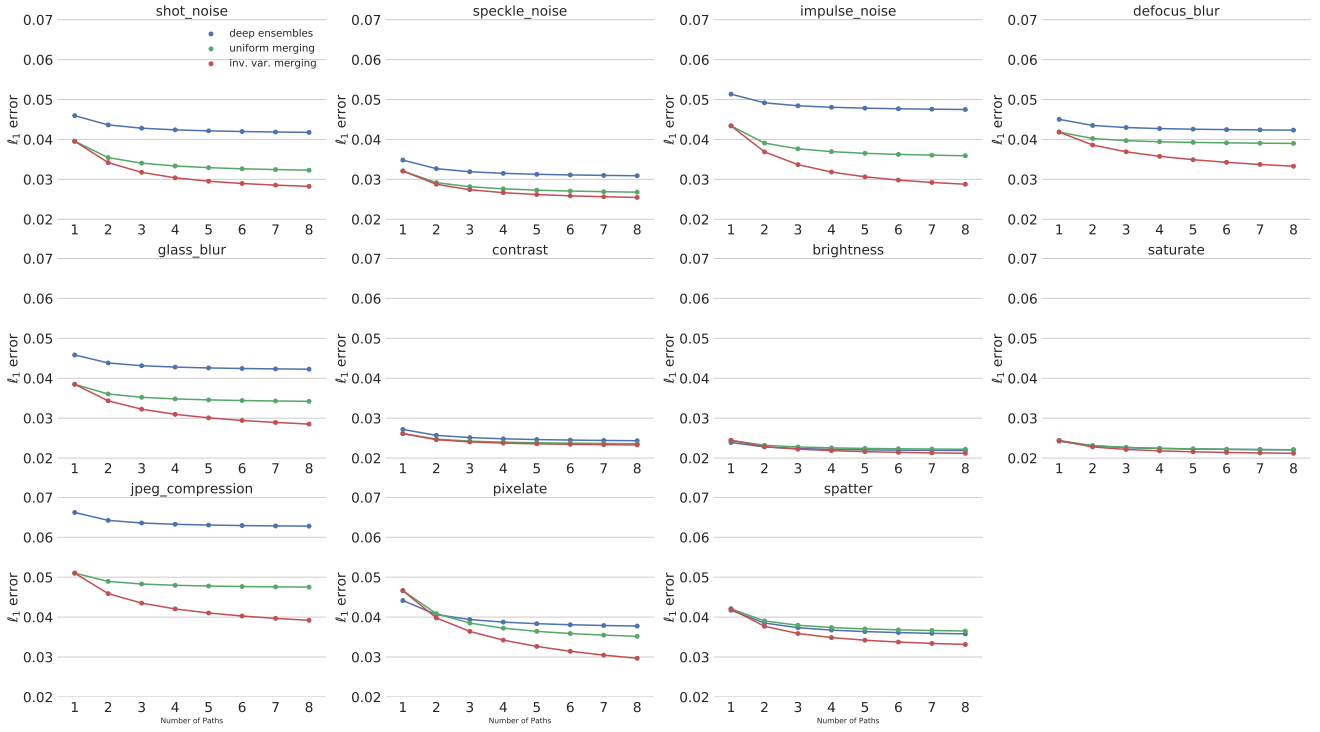


(a) Normal



(b) Reshade

Figure 10: **Robustness as a function of number of paths.** This is an extension of Fig. 7a (bottom) in the main paper and shows the ℓ_1 error for each *unseen* distortion as we increase the number of paths. For deep ensembles, this corresponds to increasing the number of ensemble components. The proposed method (*inv. var. merging*) and its simpler variants (*uniform merging*) consistently outperforms deep ensembles.



(c) Depth

Figure 10: **Robustness as a function of number of paths.** This is an extension of Fig. 7a (bottom) in the main paper and shows the ℓ_1 error for each *unseen* distortion as we increase the number of paths. For deep ensembles, this corresponds to increasing the number of ensemble components. The proposed method (*inv. var. merging*) and its simpler variants (*uniform merging*) consistently outperforms deep ensembles.

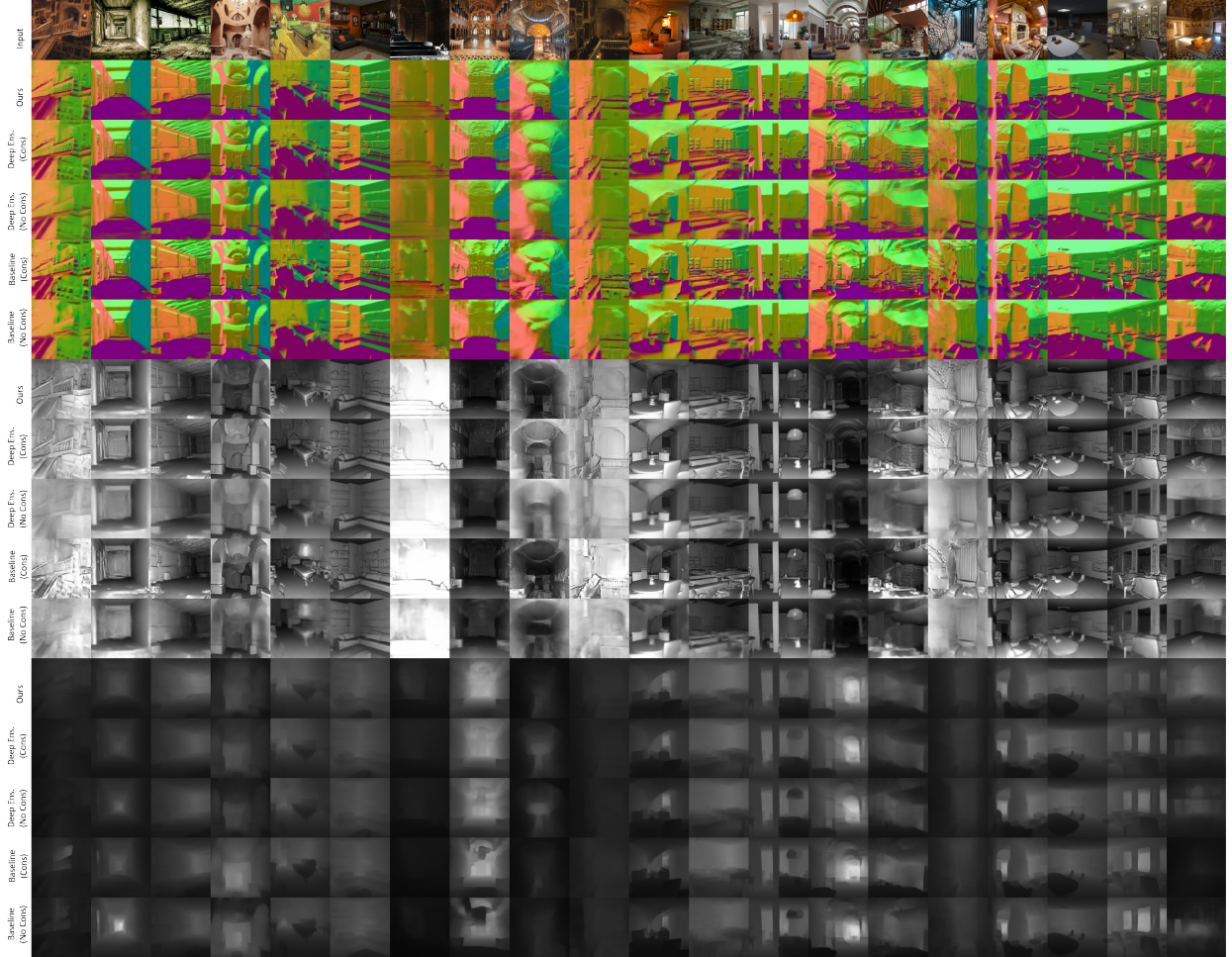


Figure 11: Qualitative results on *undistorted* external query images for the proposed method and baselines trained with (Cons) and without (No Cons) consistency constraints.

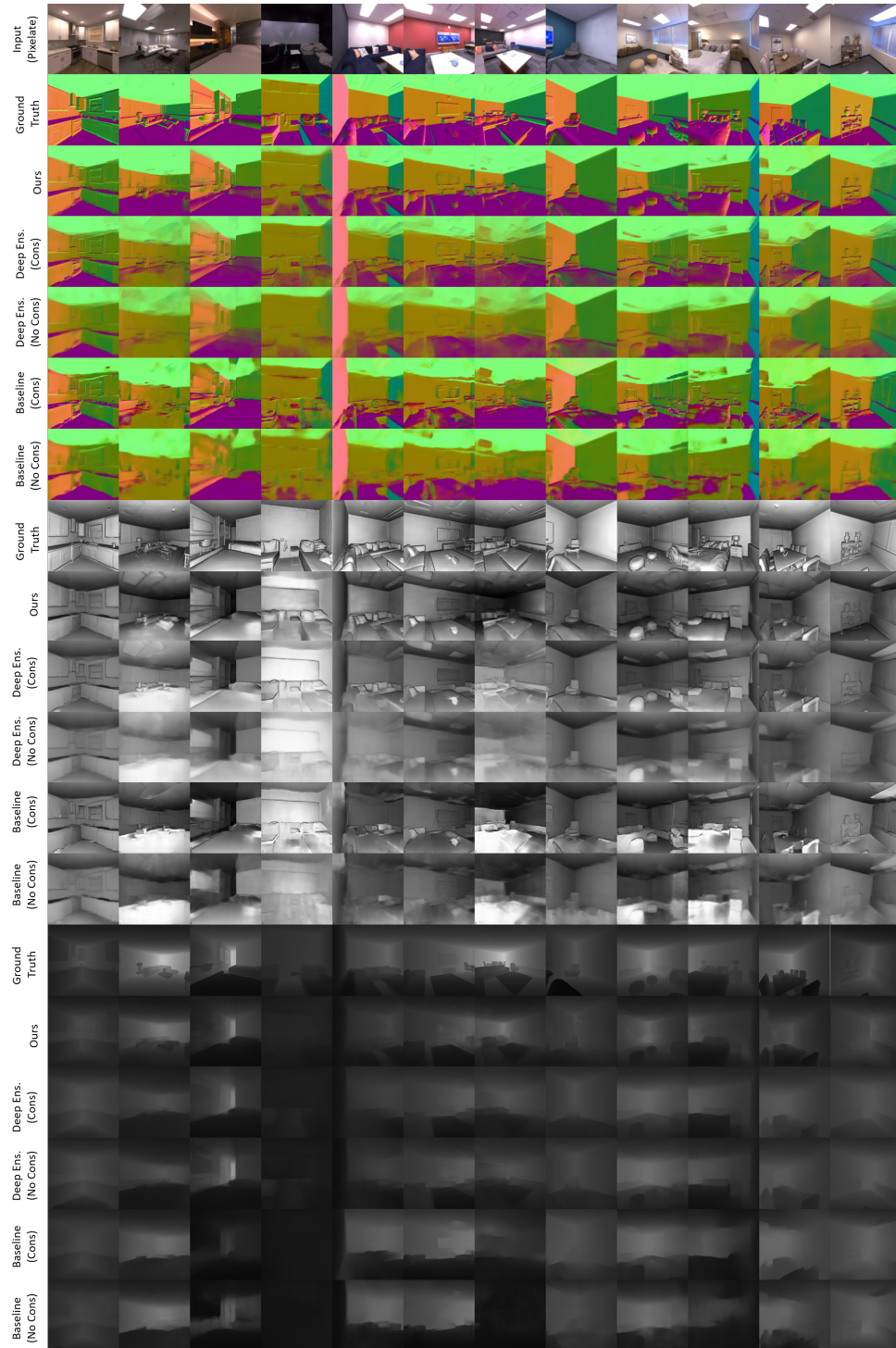


Figure 12: Qualitative results on Replica images with pixelate distortion (shift intensity 3) for the proposed method and baselines trained with (Cons) and without (No Cons) consistency constraints.

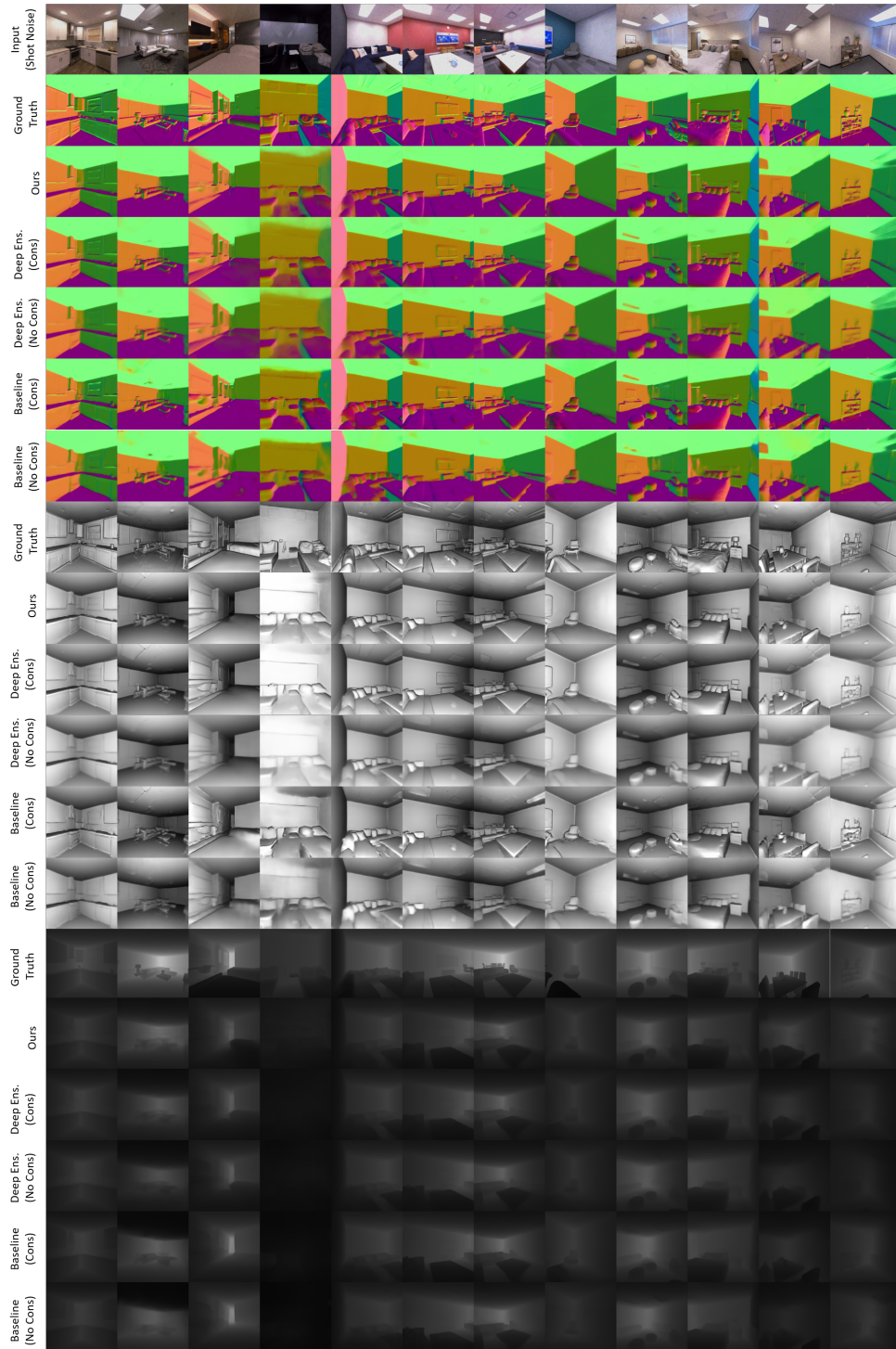


Figure 13: Qualitative results on Replica images with shot noise distortion (shift intensity 3) for the proposed method and baselines trained with (Cons) and without (No Cons) consistency constraints.

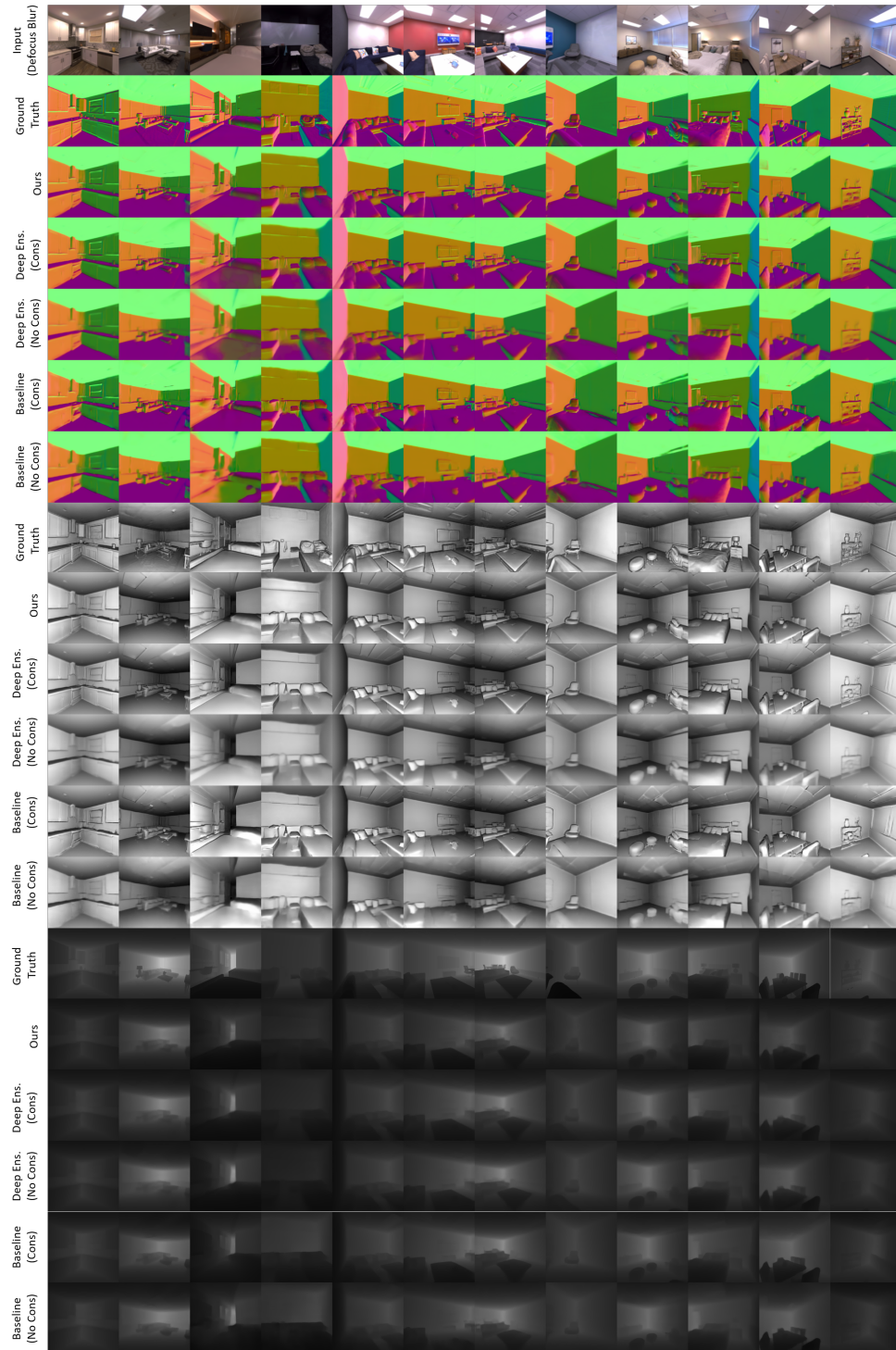


Figure 14: Qualitative results on Replica images with defocus blur distortion (shift intensity 3) for the proposed method and baselines trained with (Cons) and without (No Cons) consistency constraints.

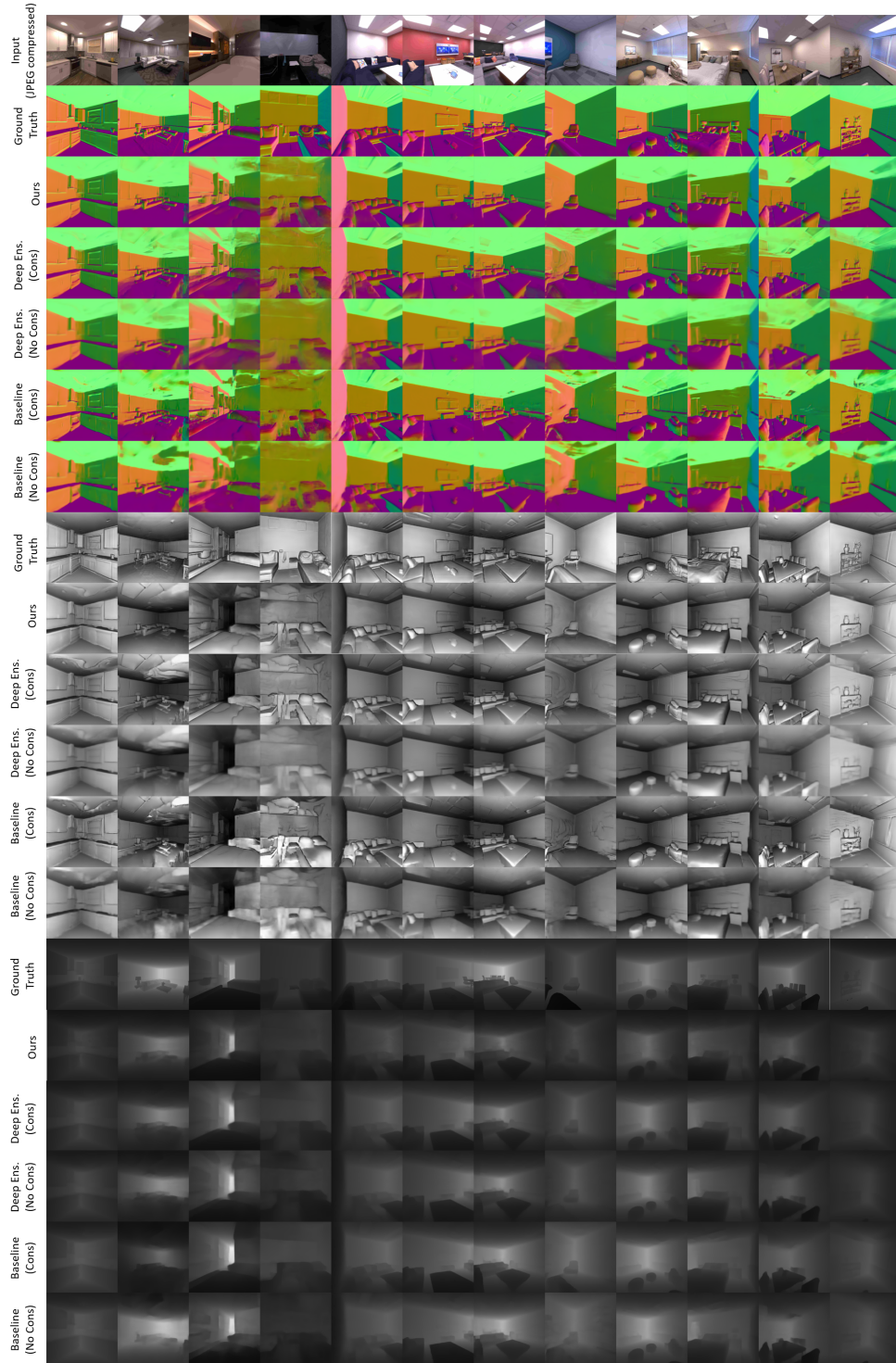


Figure 15: Qualitative results on Replica images with JPEG distortion (shift intensity 3) for the proposed method and baselines trained with (Cons) and without (No Cons) consistency constraints.

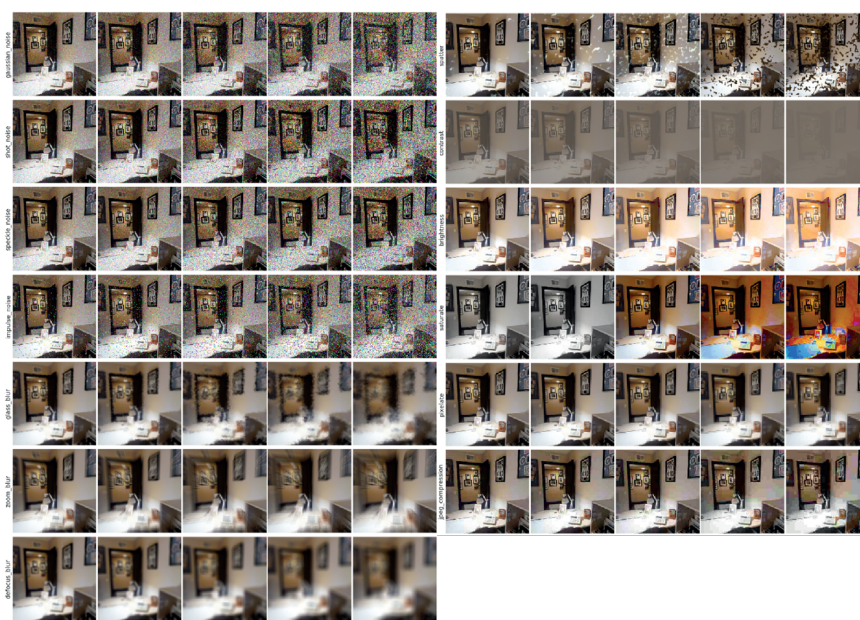


Figure 16: **Visuals of common corruptions** used for a single image sample with increasing severities from left to right.