7 Supplementary Material

7.1 Negative pull inhibition

When the inhibition parameter α is set to be trainable, the network can learn negative inhibition values (see Fig. 6). Earlier, in Fig. 5 we analyzed the characteristics of PushPullConv when the inhibition values were positive. We hereby present a similar analysis when the inhibition values are negative. In this case, the pull response is added to the push, resulting in the following computation of PushPullConv (without non-linearities):

$$\mathbf{x}_{\text{out}} = \mathbf{x} * \mathbf{w} + |\alpha| \cdot (\mathbf{x} * \mathbf{w}_{\text{pull}} * \mathbf{1}) + b \tag{14}$$

$$= \mathbf{x} * (\mathbf{w} + |\alpha| \cdot \mathbf{w}_{\text{pull}} * \mathbf{1}) + b \tag{15}$$

$$= \mathbf{x} * \tilde{f}(\mathbf{w}; \alpha, \mathbf{1}) + b \tag{16}$$

where $\mathbf{x}$ and $\mathbf{x}_{\text{out}}$ are the input and output of the PushPullConv, while $\mathbf{w}$, $\mathbf{w}_{\text{pull}}$, and $\mathbf{1}$ denote the push kernel, pull kernel, and average filter, respectively, and b denotes the bias and α the inhibition strength.

Fourier analysis of the kernels from ResNet50 (see Fig. 10) shows that with negative values of α, the push-pull kernels have a little effect over the push kernels. With the decreasing α, the central coefficient of the push-pull spectrum increases, which indicates the increasing absolute sum of the push-pull kernel weights in the spatial domain. This effect is not so prominent for kernels with $0 \leq \alpha \leq -1$ and therefore, such PushPull kernels have similar Fourier spectrum characteristics to push kernels. Fig. 6 depicts the distribution of learned α values, and it can be seen that the distribution is centred about zero with $|\alpha| \leq 1$ (for a majority of α values). When $-1 \leq \alpha \leq 0$, the PushPull has similar characteristics to the push, however, when $0 < \alpha \leq 1$ the PushPull exhibits one of the properties of a band-pass filter. The distribution of α values in Fig. 6 shows that the network is learning both kinds of filters for feature extraction.

7.2 Robustness transferability to other applications

We also studied the image retrieval problem to evaluate the robustness of the PushPull-Conv layer to applications other than image classification.

A similar scheme of hyper-parameters was used for the retrieval problem as that of the classification. For the exact details, we refer the reader to the code[1]. In this case, however, the network output is to determine a fixed-length binary hash, which we set to 64 (by setting the number of output neurons in the ConvNet to 64 following [4]). The outputs are driven to become binary by including a penalty term for quantization in the loss. Central Similarity Quantization (CSQ) [4] is used to determine the hash targets and the loss function is constructed as:

$$L = L_{\text{Hash}} + \lambda L_{\text{Quantization}} \tag{17}$$

[1] https://github.com/bgswaroop/pushpull-conv

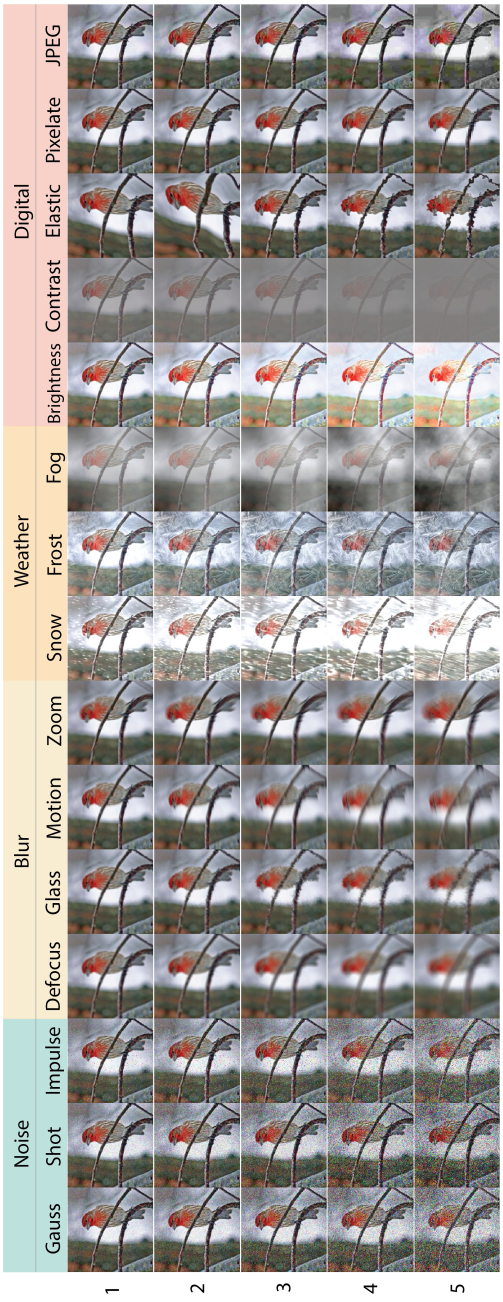

Fig. 9. The 75 different corruptions for an image taken from ImageNet-C dataset [1]. In the matrix of images, each column depicts the corruption type and each row corresponds to a specific level of severity, where 1 corresponds to the least severe and 5 the most severe. (Best viewed when zoomed to 400% in PDF).

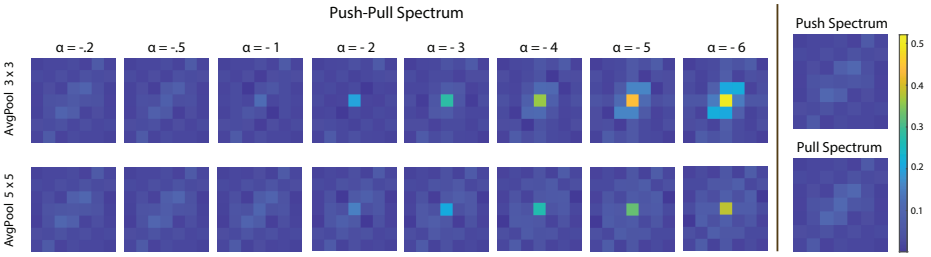

Fig. 10. Averaged Fourier spectrum determined for kernels from conv1 layer of ResNet50. The average of the Fourier spectrum of the 64 push kernels is shown on the top right (Push Spectrum). The corresponding Pull Spectrum (in the bottom right) is computed by averaging the Fourier spectrum of pull kernels (determined by Eq. (2)). The averaged Fourier transforms of a push-pull kernel as determined by $\tilde{f}$ (see Eq. (16)) are shown on the left with several configurations of the *inhibition strength* α, and averaged pooling. The actual size of each filter is $7 \times 7 \times 3$, however, this visualization is generated using filters only from the first channel. All values are scaled to a common range as shown on the right.

where L_{Hash} is the binary cross entropy loss between the true targets and the predicted hash targets $\hat{\mathbf{y}}$. The quantization weight λ is set to 1e-4 and the quantization loss for each $\hat{\mathbf{y}}$ is determined by:

$$L_{\text{Quantization}} = \frac{1}{64} \sum_{i=1}^{64} (|\tanh \hat{y}_i| - 1)^2 \tag{18}$$

Given a test query image, the trained model determines its hash. The hamming distance between the hash of the query and all the images in the training dataset is computed to determine image similarity. Accordingly, all the examples in the training dataset are ranked and the retrieval performance is determined by computing the mean average precision (mAP) for the top 200 retrieved samples. The error rate is computed as 1 - mAP.

With this setting, ResNet models were trained on the CIFAR10 dataset and evaluated for robustness on the CIFAR10-C dataset, and the results are summarized in Table 5. ResNet18 achieved *mrCE* scores of 0.98 and 0.97 while ResNet50 obtained 0.96 and 1.01 (for two configurations of pooling avg3 and avg5). Interestingly, the PushPull layer offers robustness, especially to high-frequency corruptions like Gaussian noise, shot noise, and pixelate corruptions, while not for impulse noise. Moreover, the PushPullConv offers robustness to high-frequency corruptions such as Gaussian noise, shot noise, and pixelate corruptions while not offering robustness to impulse noise. The underlying cause of this selective robustness remains unclear. While this phenomenon warrants deeper investigation, our initial studies in image retrieval suggest that the PushPullConv architecture may extend its robustness benefits to domains beyond image classification, including other real-world applications.

Table 5. Evaluation of robustness of PushPull-Net for the image retrieval problem. The ResNet models were trained on CIFAR10 and tested on CIFAR10-C datasets. E denotes the error on the clean test set (computed as 1-mAP from top-200 retrieved samples), and $mrCE$ is the mean of relative corruption errors of the 15 types of corruption. The corruption types are **Noise** - (Ga)ussian, (Sh)ot, and (Im)pulse, **Blur** - (De)foucus, (Gl)ass, (Mo)tion, and (Zo)om, **Weather** - (Sn)ow, (Fr)ost, and (Fo)g, and **Digital** - (Br)ightness, (Co)ntrast, (El)astic, (Pi)xelate, and (Jp)eg. The models ResNet18 and ResNet50 are the baseline and the two corresponding push-pull variants are shown underneath them. The scores with improved robustness are highlighted in green.

Model	E	$mrCE$	**Noise**			**Blur**				**Weather**			**Digital**				
			Ga	Sh	Im	De	Gl	Mo	Zo	Sn	Fr	Fo	Br	Co	El	Pi	Jp
ResNet18	0.071	1.000	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
PushPull avg3	0.071	0.981	0.93	0.92	1.06	0.96	1.01	0.96	0.91	0.98	0.97	1.10	1.03	1.10	0.97	0.89	0.91
PushPull avg5	0.073	0.974	0.96	0.93	1.05	0.92	1.00	0.98	0.89	0.99	0.98	1.11	1.03	1.06	0.95	0.85	0.91
ResNet50	0.071	1.000	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
PushPull avg3	0.073	0.963	0.91	0.90	1.00	0.93	0.96	0.95	0.94	1.01	0.93	0.99	1.05	1.02	0.96	0.94	0.95
PushPull avg5	0.075	1.010	0.94	0.94	1.04	1.03	0.98	1.03	1.03	1.04	0.97	1.03	1.05	1.03	1.03	1.01	0.99

7.3 Absolute scores

The results presented in Table 1, Table 2, and Table 5 show the corruption scores normalized with respect to a baseline. This scheme helps to highlight the improvements in robustness, however, it is equally important to present the absolute corruption errors for other researchers to draw comparisons with our results (as they may use different baselines). To this end, we provide these analogous results in Tables 6 to 8.

A closer look at these scores reveals that every corruption type has a different impact on the model's performance. For instance, corruption errors on IMAGENET for ResNet50 (from Table 6) reveal that all noise corruptions together with glass blur, and snow corruption have a higher degree of impact on the model's performance when compared to the rest. As demonstrated earlier, PushPull-Conv offers robustness to these corruptions, especially to high-frequency corruptions.

Table 6. Absolute scores for the image classification problem. The results presented here are analogous to the relative scores in Table 1.

D	Model	E	mCE	Noise			Blur				Weather			Digital				
				Ga	Sh	Im	De	Gl	Mo	Zo	Sn	Fr	Fo	Br	Co	El	Pi	Jp
CIFAR10	ResNet18 (b1)	0.069	0.285	0.55	0.42	0.42	0.20	0.51	0.26	0.25	0.22	0.27	0.14	0.09	0.25	0.18	0.29	0.22
	PushPull avg3	0.071	0.273	0.47	0.36	0.43	0.20	0.51	0.26	0.25	0.22	0.25	0.15	0.09	0.26	0.18	0.27	0.20
	PushPull avg5	0.071	0.271	0.48	0.38	0.41	0.20	0.47	0.25	0.25	0.22	0.26	0.14	0.09	0.26	0.18	0.27	0.20
	ResNet50 (b2)	0.069	0.303	0.56	0.43	0.40	0.23	0.57	0.28	0.29	0.24	0.31	0.15	0.09	0.24	0.20	0.33	0.22
	PushPull avg3	0.076	0.291	0.49	0.38	0.40	0.23	0.53	0.29	0.28	0.24	0.28	0.16	0.09	0.26	0.20	0.32	0.21
	PushPull avg5	0.068	0.287	0.50	0.39	0.37	0.22	0.54	0.28	0.27	0.23	0.28	0.15	0.09	0.25	0.20	0.32	0.21
IMGNET100	ResNet18 (b3)	0.207	0.581	0.78	0.77	0.81	0.71	0.65	0.58	0.56	0.67	0.63	0.58	0.31	0.66	0.36	0.35	0.31
	PushPull avg3	0.208	0.559	0.71	0.70	0.75	0.71	0.62	0.57	0.57	0.68	0.63	0.56	0.31	0.66	0.35	0.27	0.30
	PushPull avg5	0.221	0.566	0.72	0.70	0.75	0.72	0.61	0.58	0.56	0.68	0.64	0.59	0.33	0.68	0.36	0.30	0.30
	ResNet50 (b4)	0.189	0.569	0.77	0.75	0.79	0.71	0.64	0.56	0.54	0.69	0.64	0.58	0.30	0.66	0.33	0.28	0.29
	PushPull avg3	0.194	0.552	0.73	0.71	0.75	0.69	0.58	0.57	0.55	0.66	0.63	0.58	0.31	0.65	0.34	0.25	0.28
	PushPull avg5	0.203	0.545	0.70	0.69	0.72	0.67	0.56	0.54	0.52	0.69	0.63	0.61	0.32	0.68	0.33	0.26	0.27
IMGNET200	ResNet18 (b5)	0.246	0.642	0.83	0.83	0.87	0.73	0.71	0.68	0.66	0.72	0.67	0.57	0.34	0.66	0.45	0.47	0.44
	PushPull avg3	0.252	0.629	0.80	0.80	0.84	0.73	0.69	0.66	0.65	0.72	0.66	0.59	0.35	0.67	0.44	0.42	0.40
	PushPull avg5	0.253	0.628	0.81	0.81	0.84	0.73	0.69	0.66	0.66	0.71	0.66	0.58	0.35	0.67	0.44	0.41	0.40
	ResNet50 (b6)	0.213	0.618	0.82	0.82	0.85	0.69	0.68	0.65	0.62	0.70	0.65	0.55	0.31	0.64	0.42	0.45	0.41
	PushPull avg3	0.217	0.608	0.81	0.81	0.84	0.70	0.66	0.64	0.61	0.71	0.64	0.55	0.31	0.63	0.41	0.37	0.39
	PushPull avg5	0.222	0.606	0.81	0.80	0.83	0.68	0.66	0.64	0.61	0.71	0.64	0.58	0.32	0.65	0.42	0.38	0.36
IMAGENET	ResNet18 (b7)	0.325	0.717	0.84	0.84	0.88	0.74	0.79	0.72	0.75	0.78	0.73	0.65	0.42	0.71	0.61	0.67	0.62
	PushPull avg3	0.363	0.678	0.80	0.79	0.83	0.70	0.69	0.69	0.73	0.76	0.70	0.71	0.45	0.72	0.56	0.44	0.57
	PushPull avg5	0.337	0.695	0.83	0.83	0.87	0.73	0.76	0.71	0.74	0.78	0.71	0.66	0.43	0.71	0.57	0.53	0.60
	ResNet50 (b8)	0.269	0.667	0.78	0.79	0.83	0.68	0.75	0.68	0.70	0.75	0.67	0.59	0.37	0.66	0.58	0.64	0.55
	PushPull avg3	0.282	0.645	0.77	0.77	0.81	0.66	0.72	0.66	0.68	0.74	0.67	0.60	0.36	0.65	0.53	0.52	0.54
	PushPull avg5	0.276	0.645	0.77	0.77	0.82	0.66	0.71	0.66	0.68	0.73	0.65	0.58	0.36	0.65	0.54	0.54	0.55

Table 7. Absolute scores for comparison with the state-of-the-art. The results in this Table are analogous to the relative scores which were presented earlier in Table 2. BlurConv1 [5], BlurConv2 [3], PushPull1 [2]

D	Model	E	mCE	Noise			Blur				Weather			Digital				
				Ga	Sh	Im	De	Gl	Mo	Zo	Sn	Fr	Fo	Br	Co	El	Pi	Jp
CIFAR10	ResNet50 (b2)	0.069	0.281	0.50	0.39	0.36	0.21	0.52	0.28	0.26	0.22	0.27	0.16	0.09	0.28	0.19	0.28	0.20
	PushPull avg3 (ours)	0.076	0.268	0.45	0.35	0.36	0.20	0.50	0.26	0.25	0.22	0.25	0.16	0.09	0.29	0.18	0.27	0.19
	PushPull avg5 (ours)	0.068	0.281	0.47	0.37	0.37	0.22	0.51	0.29	0.27	0.23	0.26	0.17	0.10	0.29	0.19	0.28	0.19
	PushPull1	0.069	0.298	0.51	0.40	0.38	0.23	0.55	0.29	0.28	0.26	0.31	0.17	0.10	0.25	0.20	0.34	0.21
	BlurConv1	0.069	0.283	0.47	0.38	0.40	0.22	0.50	0.27	0.29	0.23	0.29	0.15	0.08	0.23	0.20	0.32	0.22
	BlurConv2	0.098	0.365	0.59	0.48	0.43	0.30	0.64	0.41	0.39	0.31	0.35	0.21	0.12	0.33	0.26	0.41	0.24
IMAGENET	ResNet50 (b8)	0.269	0.667	0.78	0.79	0.83	0.68	0.75	0.68	0.70	0.75	0.67	0.59	0.37	0.66	0.58	0.64	0.55
	PushPull avg3 (PP3)	0.282	0.645	0.77	0.77	0.81	0.66	0.72	0.66	0.68	0.74	0.67	0.60	0.36	0.65	0.53	0.52	0.54
	PushPull avg5 (PP5)	0.276	0.645	0.77	0.77	0.82	0.66	0.71	0.66	0.68	0.73	0.65	0.58	0.36	0.65	0.54	0.54	0.55
	PushPull1	0.269	0.665	0.78	0.79	0.84	0.67	0.75	0.67	0.69	0.74	0.66	0.59	0.36	0.66	0.58	0.63	0.57
	BlurConv1	0.276	0.626	0.70	0.70	0.73	0.67	0.72	0.67	0.68	0.74	0.65	0.58	0.37	0.66	0.57	0.45	0.48
	BlurConv2	0.265	0.621	0.74	0.74	0.79	0.64	0.70	0.65	0.66	0.72	0.64	0.57	0.35	0.64	0.50	0.49	0.51
	BlurConv2+PP3	0.307	0.617	0.71	0.71	0.74	0.64	0.67	0.65	0.65	0.75	0.65	0.64	0.39	0.68	0.53	0.36	0.48
	Res50+AugMix (b9)	0.269	0.612	0.71	0.72	0.74	0.61	0.75	0.59	0.60	0.69	0.63	0.49	0.35	0.59	0.58	0.63	0.51
	Res50+AugMix+PP3	0.287	0.590	0.67	0.67	0.69	0.59	0.70	0.58	0.57	0.70	0.63	0.52	0.36	0.60	0.52	0.52	0.52
	Res50+AutoAug (b10)	0.269	0.630	0.70	0.71	0.75	0.69	0.71	0.68	0.73	0.71	0.64	0.51	0.33	0.50	0.62	0.65	0.53
	Res50+AutoAug+PP3	0.283	0.604	0.67	0.67	0.70	0.67	0.68	0.68	0.71	0.71	0.63	0.52	0.34	0.47	0.58	0.53	0.51
	Res50+PRIME (b11)	0.298	0.522	0.48	0.49	0.48	0.60	0.65	0.58	0.60	0.60	0.55	0.52	0.35	0.49	0.51	0.47	0.49
	Res50+PRIME+PP3	0.306	0.500	0.44	0.44	0.43	0.59	0.63	0.57	0.61	0.60	0.54	0.52	0.35	0.47	0.49	0.40	0.44
	Res50+TrivialAug (b12)	0.269	0.596	0.66	0.67	0.71	0.65	0.69	0.63	0.69	0.67	0.63	0.48	0.33	0.47	0.57	0.60	0.49
	Res50+TrivialAug+PP3	0.286	0.587	0.67	0.68	0.70	0.65	0.66	0.63	0.68	0.67	0.63	0.52	0.34	0.46	0.52	0.49	0.48
	EffNet (b13)	0.286	0.682	0.81	0.83	0.84	0.71	0.77	0.66	0.72	0.72	0.68	0.59	0.38	0.70	0.58	0.65	0.61
	EffNet+PP3	0.288	0.666	0.80	0.81	0.82	0.70	0.75	0.66	0.72	0.72	0.68	0.59	0.38	0.70	0.57	0.59	0.52

Table 8. Absolute scores for the image retrieval problem on the CIFAR10-C dataset. The results in this Table are analogous to the relative scores which were presented earlier in Table 5.

Model	E	mCE	Noise			Blur				Weather			Digital				
			Ga	Sh	Im	De	Gl	Mo	Zo	Sn	Fr	Fo	Br	Co	El	Pi	Jp
ResNet18	0.071	0.264	0.47	0.36	0.39	0.19	0.47	0.25	0.24	0.20	0.22	0.14	0.09	0.27	0.18	0.26	0.20
PushPull avg3	0.071	0.258	0.44	0.33	0.42	0.19	0.48	0.24	0.22	0.20	0.22	0.15	0.09	0.30	0.18	0.23	0.18
PushPull avg5	0.073	0.256	0.45	0.34	0.41	0.18	0.47	0.25	0.22	0.20	0.22	0.16	0.09	0.29	0.17	0.22	0.19
ResNet50	0.071	0.281	0.50	0.39	0.36	0.21	0.52	0.28	0.26	0.22	0.27	0.16	0.09	0.28	0.19	0.28	0.20
PushPull avg3	0.073	0.268	0.45	0.35	0.36	0.20	0.50	0.26	0.25	0.22	0.25	0.16	0.09	0.29	0.18	0.27	0.19
PushPull avg5	0.075	0.281	0.47	0.37	0.37	0.22	0.51	0.29	0.27	0.23	0.26	0.17	0.10	0.29	0.19	0.28	0.19

References

1. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. In: International Conference on Learning Representations (2019)
2. Strisciuglio, N., Lopez-Antequera, M., Petkov, N.: Enhanced robustness of convolutional networks with a push–pull inhibition layer. Neural Computing and Applications **32**(24), 17957–17971 (2020)
3. Vasconcelos, C., Larochelle, H., Dumoulin, V., Roux, N.L., Goroshin, R.: An effective anti-aliasing approach for residual networks. arXiv preprint arXiv:2011.10675 (2020)
4. Yuan, L., Wang, T., Zhang, X., Tay, F.E., Jie, Z., Liu, W., Feng, J.: Central similarity quantization for efficient image and video retrieval. In: CVPR. pp. 3083–3092 (2020)
5. Zhang, R.: Making convolutional networks shift-invariant again. In: International conference on machine learning. pp. 7324–7334. PMLR (2019)