Supplementary material for "SODA: Robust Training of Test-Time Data Adaptors"

A Derivation of Directional Derivative Approximation in SODA

In Section 3.2, given a deployed model $\mathbf{M}$, the ideal objective function of training the data adaptor $\mathbf{G}$ with parameters $\boldsymbol{\theta}$ in SODA is the KL divergence between the predicted probability $\hat{\mathbf{p}}_i^{\boldsymbol{\theta}} = \mathbf{M} \circ \mathbf{G}(\mathbf{x}_i; \boldsymbol{\theta})$ of the adapted data point $\mathbf{x}_i^{\boldsymbol{\theta}} = \mathbf{G}(\mathbf{x}_i; \boldsymbol{\theta})$ and the true label $\mathbf{y}_i$ of the original data point $\mathbf{x}_i$. Because $\mathbf{y}_i$ is not available at test time, pseudo-label $\hat{\mathbf{y}}_i$ predicted by $\mathbf{M}$ is adopted as a substitute of $\mathbf{y}_i$. Due to the inaccurate model prediction under distribution shifts, there is a disturbance $\boldsymbol{\sigma}_i$ in $\hat{\mathbf{y}}_i$ compared to $\mathbf{y}_i$, i.e. $\hat{\mathbf{y}}_i = \boldsymbol{\sigma}_i + \mathbf{y}_i$. Hence, the KL divergence loss $\mathcal{L}(\cdot, \cdot) := \mathrm{KL}(\cdot \| \cdot)$ at test data point $\mathbf{x}_i$ is

$$\begin{aligned}
\mathcal{L}_i &= KL(\hat{\mathbf{y}}_i \| \hat{\mathbf{p}}_i^{\boldsymbol{\theta}}) = \hat{\mathbf{y}}_i \log \frac{\hat{\mathbf{y}}_i}{\hat{\mathbf{p}}_i^{\boldsymbol{\theta}}} \\
&= (\mathbf{y}_i + \boldsymbol{\sigma}_i) \log \frac{\mathbf{y}_i + \boldsymbol{\sigma}_i}{\hat{\mathbf{p}}_i^{\boldsymbol{\theta}}} \\
&= (\mathbf{y}_i + \boldsymbol{\sigma}_i) \log(\mathbf{y}_i + \boldsymbol{\sigma}_i) - \mathbf{y}_i \log \hat{\mathbf{p}}_i^{\boldsymbol{\theta}} - \boldsymbol{\sigma}_i \log \hat{\mathbf{p}}_i^{\boldsymbol{\theta}} \\
&= -H(\mathbf{y}_i + \boldsymbol{\sigma}_i) + \mathcal{L}_{\mathrm{ce}}(\mathbf{y}_i, \hat{\mathbf{p}}_i^{\boldsymbol{\theta}}) - \boldsymbol{\sigma}_i \log \hat{\mathbf{p}}_i^{\boldsymbol{\theta}}
\end{aligned} \tag{9}$$

where $\mathcal{L}_{\mathrm{ce}}(\mathbf{y}_i, \hat{\mathbf{p}}_i^{\boldsymbol{\theta}})$ is the cross entropy loss between $\mathbf{y}_i$ and $\hat{\mathbf{p}}_i^{\boldsymbol{\theta}}$. Because the gradient information is inaccessible from the deployed model, zeroth-order optimization (ZOO) is utilized to estimate gradients for the training of data adaptor in SODA. To do this, the objective function $f(\boldsymbol{\theta})$ in Eq. (1) is replaced with the training objective function $\mathcal{L}_i$ in test-time data adaptation. Denote $\mathcal{L}_i^{\boldsymbol{\theta}}$ as the KL divergence loss computed by data adaptor with parameters $\boldsymbol{\theta}$, the directional derivative approximation of ZOO is

$$\begin{aligned}
\widehat{\nabla}_{\boldsymbol{\theta}} \check{\mathcal{L}}_i &= \frac{1}{\mu q} \sum_{j=1}^{q} \left[KL(\hat{\mathbf{y}}_i \| \hat{\mathbf{p}}_i^{\boldsymbol{\theta} + \boldsymbol{\mu} \mathbf{u}_j}) - KL(\hat{\mathbf{y}}_i \| \hat{\mathbf{p}}_i^{\boldsymbol{\theta}}) \right] \\
&= \frac{1}{\mu q} \sum_{j=1}^{q} \left[\left(\mathcal{L}_{\mathrm{ce}}(\mathbf{y}, \hat{\mathbf{p}}_i^{\boldsymbol{\theta} + \mu \mathbf{u}_j}) - \boldsymbol{\sigma}_i \log \hat{\mathbf{p}}_i^{\boldsymbol{\theta} + \mu \mathbf{u}_j} \right) - \left(\mathcal{L}_{\mathrm{ce}}(\mathbf{y}, \hat{\mathbf{p}}_i^{\boldsymbol{\theta}}) - \boldsymbol{\sigma}_i \log \hat{\mathbf{p}}_i^{\boldsymbol{\theta}} \right) \right] \\
&= \frac{1}{\mu q} \sum_{j=1}^{q} \left[\mathcal{L}_{\mathrm{ce}}(\mathbf{y}, \hat{\mathbf{p}}_i^{\boldsymbol{\theta} + \mu \mathbf{u}_j}) - \mathcal{L}_{\mathrm{ce}}(\mathbf{y}, \hat{\mathbf{p}}_i^{\boldsymbol{\theta}}) \right] + \frac{1}{\mu q} \sum_{j=1}^{q} \left[\boldsymbol{\sigma}_i \log \hat{\mathbf{p}}_i^{\boldsymbol{\theta}} - \boldsymbol{\sigma}_i \log \hat{\mathbf{p}}_i^{\boldsymbol{\theta} + \mu \mathbf{u}_j} \right] \\
&= \widehat{\nabla}_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{ce}} + \frac{\boldsymbol{\sigma}_i}{\mu q} \sum_{j=1}^{q} \log \frac{\hat{\mathbf{p}}_i^{\boldsymbol{\theta}}}{\hat{\mathbf{p}}_i^{\boldsymbol{\theta} + \mu \mathbf{u}_j}},
\end{aligned} \tag{10}$$

where $\widehat{\nabla}_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{ce}} = \frac{1}{\mu q} \sum_{j=1}^{q} \left[\mathcal{L}_{\mathrm{ce}}(\mathbf{y}_i, \hat{\mathbf{p}}_i^{\boldsymbol{\theta} + \mu \mathbf{u}_j}) - \mathcal{L}_{\mathrm{ce}}(\mathbf{y}_i, \hat{\mathbf{p}}_i^{\boldsymbol{\theta}}) \right]$ is the ideal directional derivative approximation.

B Implementation Details

B.1 Implementation details of DINE and BETA

The implementations of DINE and BETA on CIFAR-10-C and CIFAR-100-C are kept the same, following their original work [18] and [41]. For DINE, the momentum hyper-parameter γ = 0.6, and the Mixup balancing hyper-parameter β = 1. For BETA, τ = 0.8 for domain division, α = 1.0 for Mixup, λ_{mse} = 0, sharpening factor T = 0.5, and adversarial regularier γ = 0.1. The training strategy of DINE and BETA are both SGD with learning rate = 0.001 for target network backbones and 0.01 for MLP classifiers. momentum = 0.9 and weight decay = 1e-3 are also adopted.

B.2 Software and hardware

In our paper, all models are implemented using PyTorch 1.13.1. The ImageNet pre-trained weights used in DINE and BETA is downloaded from TorchVision 0.14.1. The experiments are conducted using an NVIDIA A100-PCIE-40GB GPU with CUDA 11.7.

B.3 Network structure of data adaptor

Figure 5 shows the network structure of data adaptor used in our experiments. The basic structure of the data adaptor consists of two convolutional layers and an instance normalization layer in-between. Multiple ResNet blocks can be inserted into the convolutional layers to form a deeper network as in [29]. For all methods except DA-Direct, the adapted data is generated by treating the network output as perturbation and adding it to the original data.

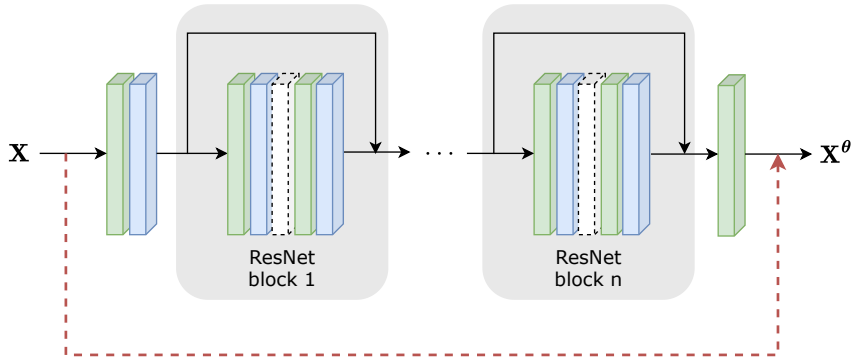


Figure 5: Network structure of data adaptor. The green block is convolutional layer, the blue block is instance normalization layer and the dashed white block is dropout layer which can be removed. The red dashed line means the network output is added to the original data to generate the adapted data.

C Additional Analysis

C.1 Discussion about SODA and SODA-R

Compared with SODA, SODA-R not only uses computed first-order gradients, but also adopts several techniques to improve the performance, i.e. deeper data adaptor with 2 ResNet blocks, Adam optimizer, perturbation regularization and dropout. The effect of network complexity has already been discussed in Section 4.3. In this subsection, we first introduce the perturbation regularization used in SODA-R, then evaluate the effect of perturbation regularization, different optimizers and dropout on SODA and SODA-R.

C.1.1 Perturbation regularization in SODA-R

In SODA and SODA-R, the adapted data is computed by perturbing the original data with a generated perturbation. To further restrict the impact of generated perturbations on data $\mathbf{X}_r = \{\mathbf{x}_{r_1}, \mathbf{x}_{r_2}, ..., \mathbf{x}_{r_{l_r}}\}$ and $\mathbf{X}_u = \{\mathbf{x}_{r_1}, \mathbf{x}_{r_2}, ..., \mathbf{x}_{r_{l_u}}\}$, perturbation regularization with l_1 norm is used: let $\mathbf{x}_{r_i}^{\boldsymbol{\theta}}$ and $\mathbf{x}_{u_i}^{\boldsymbol{\theta}}$ be the corresponding adapted data of $\mathbf{x}_{r_i}$ and $\mathbf{x}_{r_u}$,

$$\mathcal{R}(\mathbf{X}) = \mathbb{E}_{\mathbf{x}_i \in \mathbf{X}_r} \left\| \mathbf{x}_{r_i}^{\boldsymbol{\theta}} - \mathbf{x}_{r_i} \right\|_1 + \mathbb{E}_{\mathbf{x}_i \in \mathbf{X}_u} \left\| \mathbf{x}_{u_i}^{\boldsymbol{\theta}} - \mathbf{x}_{u_i} \right\|_1. \tag{11}$$

First-order gradients of the perturbation regularization is directly computed and back-propagated through the data adaptor. Hence, the training objective of SODA-R becomes:

$$\mathcal{L}_{\text{all}}(\mathbf{X}, \hat{\mathbf{Y}}_r) = \mathcal{L}_{\text{im}}(\mathbf{X}_u) + \alpha \mathcal{L}_{\text{ce}}(\mathbf{X}_r, \hat{\mathbf{Y}}_r) + \beta \mathcal{R}(\mathbf{X}), \tag{12}$$

where β is the weight of perturbation regularization and set to be 0.005 for CIFAR-10-C and 0.01 for CIFAR-100-C.

C.1.2 Evaluation of perturbation regularization in SODA and SODA-R

We evaluate the effect of perturbation regularization in SODA and SODA-R on CIFAR-10-C and CIFAR-100-C. Except for perturbation regularization term in training objective, all other settings are kept the same as in the main experiments. The results are shown in Table 6 and Table 7. It shows that perturbation regularization can improve the performance of SODA-R using first-order optimization,

especially on CIFAR-100-C. However, it largely hinders the performance of SODA using zeroth-order optimization. The computed first-order gradients of perturbation regularization is more accurate than the estimated zeroth-order gradients of the main training objective. Thus, the data adaptor tends to optimize the perturbation regularization term first, resulting in perturbations with too small norms. The perturbations with too small norms do not have enough ability to modify the test data, which might be the reason of worse performance achieved by SODA with perturbation regularization. One possible solution to solve this problem could be treating perturbation regularization as an optimization constraint, and using constrained ZOO methods to train the data adaptor.

Table 6: Comparing of SODA and SODA-R with and without perturbation regularization on CIFAR-10-C. $\beta = 0.005$ in experiments w/ regularization.

Methods	SODA	SODA-R
w/ regularization	73.40	88.28
w/o regularization	82.54	87.96

Table 7: Comparing of SODA and SODA-R with and without perturbation regularization on CIFAR-100-C. $\beta = 0.01$ in experiments w/ regularization.

Methods	SODA	SODA-R
w/ regularization	42.27	60.26
w/o regularization	52.51	58.11

C.1.3 Evaluation of optimizers in SODA and SODA-R

We evaluate the effect of optimizers used in SODA and SODA-R on CIFAR-10-C and CIFAR-100-C. Except for the optimizer used to train the data adaptor, all other settings are kept the same as in the main experiments. The results are shown in Table 8 and Table 9. On CIFAR-10-C, SODA trained by SGD and Adam achieve almost the same accuracy, while SODA-R trained by Adam achieves 3.3% higher accuracy than SODA-R trained by SGD. On CIFAR-100-C, SODA-R trained by Adam still outperforms SODA-R trained by SGD, but SODA trained by Adam achieves even worse accuracy than SODA trained by SGD. It shows that Adam optimizer has the ability to improve the training of data adaptor using first-order gradients, but fails when using the estimated zeroth-order gradients.

Table 8: Comparing of SODA and SODA-R using SGD and Adam optimizer on CIFAR-10-C.

Methods	SODA	SODA-R
SGD	82.54	84.95
Adam	82.75	88.28

Table 9: Comparing of SODA and SODA-R using SGD and Adam optimizer on CIFAR-100-C.

Methods	SODA	SODA-R
SGD	52.51	58.32
Adam	49.75	60.26

C.1.4 Evaluation of dropout in SODA and SODA-R

We also evaluate the effect of dropout on SODA and SODA-R. As depicted in Figure 5, a dropout layer can be inserted into the ResNet block. We conduct experiments using data adaptor with and without dropout layers for SODA and SODA-R. To keep the same network structure with SODA-R, the data adaptor used in SODA also has 2 ResNet blocks. The dropout ratio is set to be 0.5. All other settings are kept the same as in the main experiments. Table 10 and Table 11 show the results on CIFAR-10-C and CIFAR-100-C respectively. For SODA-R, adding dropout layers can improve the accuracy by 0.7% on CIFAR-10-C and 2% on CIFAR-100-C. However, for SODA, adding dropout layers extremely hinders the performance, especially on CIFAR-100-C. This contrast indicates that dropout has negative effect on data adaptor optimized using estimated zeroth-order gradients, while having positive effect on data adaptor optimized using computed first-order gradients. The reason might be that the extra randomness introduced by dropout increases the difficulty of gradient estimation in zeroth-order optimization. Note that, the accuracy of SODA using data adaptor with 2 ResNet blocks on CIFAR-100-C is worse than that using data adaptor with 0 ResNet blocks, which is consistent with the results on CIFAR-10-C as shown in Table 3.

Table 10: Comparing of SODA and SODA-R with and without dropout layers on CIFAR-10-C.

Methods	SODA	SODA-R
w/ dropout	32.19	88.28
w/o dropout	80.56	87.54

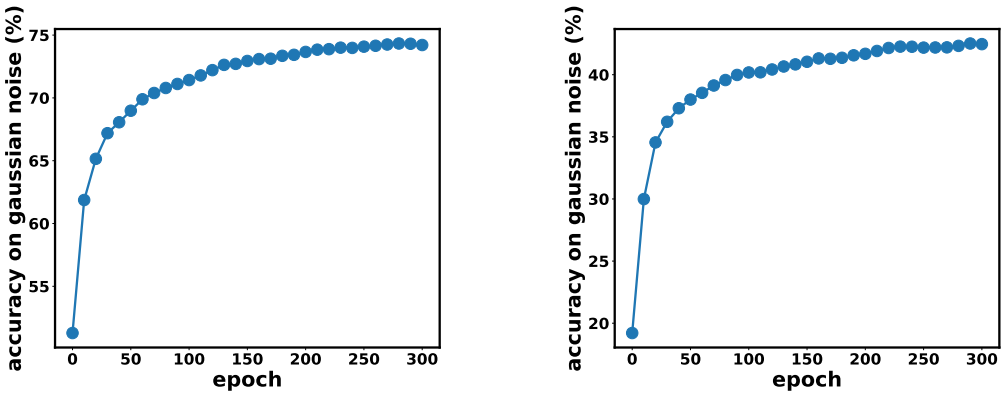
Table 11: Comparing of SODA and SODA-R with and without dropout layers on CIFAR-100-C.

Methods	SODA	SODA-R
w/ dropout	43.96	60.26
w/o dropout	5.47	58.27

To sum up, compared with SODA using zeroth-order optimization, SODA-R uses first-order optimization and adopts deeper network structure, perturbation regularization, Adam optimizer and dropout to improve the performance. However, these techniques cannot make improvement or even hinder the performance of SODA. This comparison shows that the common boosting strategies used in first-order optimization cannot be directly applied to zeroth-order optimization, leading to the limited performance of methods using zeroth-order optimization.

C.2 Convergence of SODA

In Figure 3, the convergence speeds of SODA on CIFAR-10-C and CIFAR-100-C are slower than SODA-FO and SODA-R, and do not achieve complete convergence after training with 150 epochs. We further train SODA on Gaussian noise in CIFAR-10-C and CIFAR-100-C for 300 epochs to show the complete convergence of SODA as depicted in Figure 6. With more training epochs, SODA can achieve higher accuracies on both datasets. However, training with more epochs means more adaptation processing time or more computing resources with parallel computation. For time and resource efficiency, we only report the accuracies achieved at 150 epochs in our main experiments which already improves the deployed model by a large margin. If computing time and resources are not restricted, SODA has the ability to further improve the deployed model to have higher accuracy.



(a) (b)

Figure 6: Accuracy convergence on Gaussian noise in (a) CIFAR-10-C and (b) CIFAR-100-C for 300 epochs.

C.3 Hyper-parameter analysis of reliable pseudo-label selection

We evaluate the hyper-parameters in reliable pseudo-label selection, namely the confidence threshold τ, the noise ratio ρ, and the balancing parameter α. τ and ρ controls the number of selected reliable pseudo-labels. With lower τ and lower ρ, the number of selected reliable pseudo-labels increase. We evaluate τ in $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$, and ρ in $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. Note that, when $\tau = 0$, the number of selected pseudo-labels is not equal to $(1 - \rho)n$, where n is the total number of test data points, because the pseudo-labels are not evenly distributed across classes as depicted in Figure 8a. The inaccurate model prediction tends to bias towards few classes, leading to more pseudo-labels belonging to those classes. α controls the balance between the supervised training

objective $\mathcal{L}_{\text{ce}}$ and the unsupervised training objective $\mathcal{L}_{\text{im}}$. We evaluate α in $\{0.01, 0.001, 0.0001\}$. The results of SODA with different set of hyper-paramters on CIFAR-10-C Gaussian noise level 5 corruption are showed in Figure 7. The performance of SODA is stable across different hyper-parameter settings. A common trend among different α is that accuracy tends to increase when τ and ρ decreases, i.e. the top-right corner of each figure. This trend shows that the performance of the data adaptor can be improved using more selected pseudo-labels, which further indicates the reliability of the selected pseudo-labels. There is a mild tendency of performance drop in the overall performance of SODA when unsupervised learning of test points with unreliable pseudo-labels is overwhelmed by supervised learning of reliable pseudo-labels with larger α, indicating that learning on test points with unreliable pseudo-labels also has contribution to the performance of SODA. To balancing the supervised leanring term and unsupervised learning term, we finally choose $\alpha = 0.0001$ in our main experiments. Although better performance can be achieved by carefully fine-tuning τ and ρ with a validation set, to show the general performance of SODA and select the most reliable pseudo-labels for different corruptions in both CIFAR-10-C and CIFAR-100-C datasets, we set $\tau = 0.9$ and $\rho = 0.9$ in our main experiments without elaborated hyper-parameter fine-tuning.

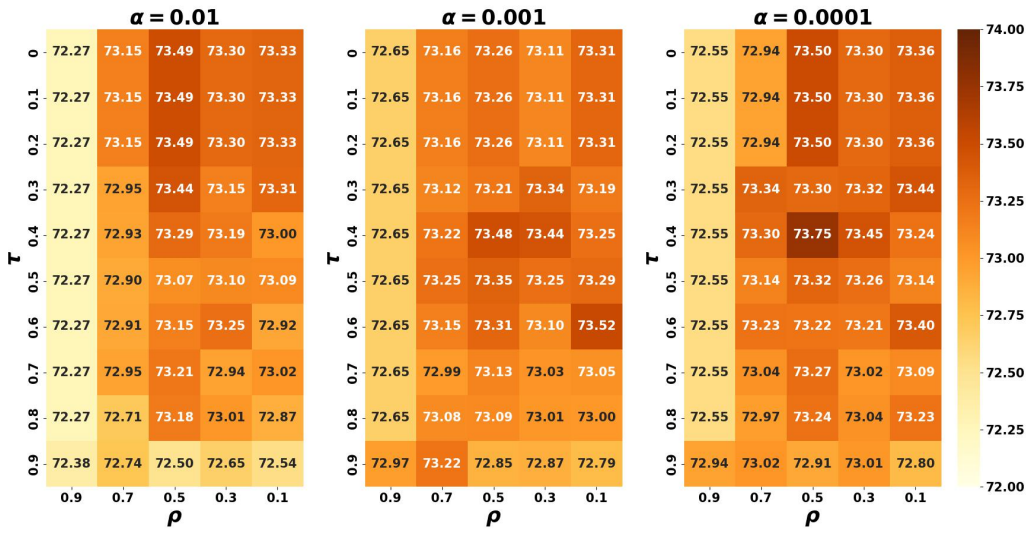


Figure 7: Evaluation of reliable pseudo-label selecting hyper-paramters on CIFAR-10-C Gaussian noise corruption level 5. Numbers are prediction accuracies (%) after adaptation.

C.4 Evaluation of queue size in SODA-O

We evaluate the effect of queue size in SODA-O. Larger queue size means more past reliable pseudo-labels and their corresponding test data points are stored and used in the adaptation process of the current mini-batch. Fixing batch size = 128, we conduct experiments on queue size $\{500, 1000, 2000, 3000\}$, and the results are shown in Table 12. The performance of SODA-O is stable with different queue sizes, especially when queue size is smaller. When queue size increases, the ratio of reliable pseudo-labels used to train the data adaptor for the current mini-batch also increases. It makes the training of the data adaptor more biased towards the supervised training with the reliable pseudo-labels. Thus, the mild performance drop observed along with larger queue size might indicate that the reliable pseudo-labels still have disturbance, and the unsupervised training of data points with unreliable pseudo-labels is useful to alleviate the negative effect caused by the remaining disturbance.

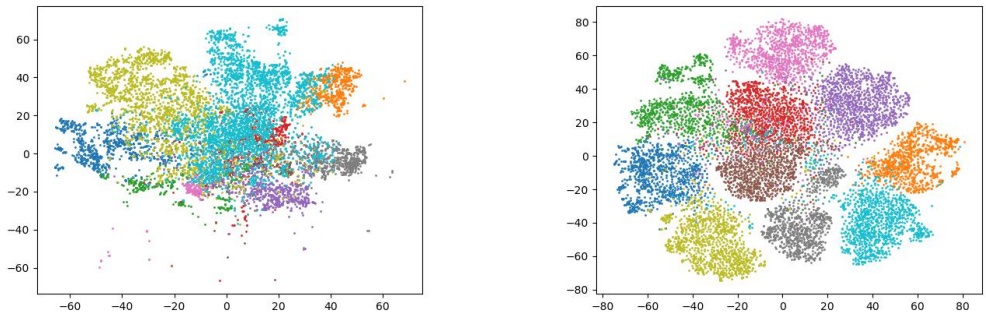
Table 12: Comparing of SODA-O with different queue sizes. Averaged accuracies (%) over 19 corruptions are reported.

Queue Size	500	1000	2000	3000
CIFAR-10-C	78.78	78.79	78.22	77.73
CIFAR-100-C	47.18	47.21	46.67	45.63

D Qualitative Evaluation of SODA

D.1 T-SNE Visuailization of SODA features

To qualitatively evaluate the performance of SODA, we use T-SNE to visualize the feature embeddings of SODA, i.e. the input features of the last classification layer in the deployed model before and after adaptation in Figure 8. According to the visualization results, the feature embeddings are much more separated apart between classes after adaptation, showing the effectiveness of SODA.



(a) Before adaptation

(b) After adaptation

Figure 8: T-SNE visualization of SODA feature embeddings on CIFAR-10-C pixelate corruption level 5.

D.2 Examples of Adapted Data

Figure 9 shows examples of test data and adapted data using SODA for 19 corruptions in CIFAR-10-C. Comparing original data without corruption, test data before adaptation and adapted data after adaptation, it is obvious that the adapted data look closer to the original data than the corresponding test data. This observation is consistent with the improved prediction accuracy using SODA, and further illustrates that the distribution shifts between the test data and the training data are alleviated after applying SODA to test data. It also indicates that SODA adapts the test data to the deployed model by modifying them to "look like" the training data. Then, the distribution shifts between the test data and the training data are mitigated, leading to improved prediction of the deployed model.

E Detailed Results

There are 19 corruptions in CIFAR-10-C and CIFAR-100-C: Gaussian noise (GN), shot noise (ShN), impulse noise (IN), speckle noise (SpN), defocus blur (DB), glass blur (GlB), motion blur (MB), zoom blur (ZB), Gaussian blur (GaB), snow (SW), frost (FR), fog (FG), brightness (BR), contrast (CT), elastic transform (ET), pixelate (PX), jpeg compression (JC), spatter (SP) and saturate (SA). We report the accuracies of each methods w.r.t. each corruption on CIFAR-10-C and CIFAR-100-C in Table 13 and Table 14. Except SODA-R and MA-SO using first order gradient from the deployed model, SODA outperforms all baselines amongst all corruptions. On CIFAR-10-C, SODA-R even outperforms MA-SO on all corruptions. On CIFAR-100-C, although the average accuracy of SODA-R is lower than the average accuracy of MA-SO, SODA-R still outperforms MA-SO on 7 corruptions.

Table 13: Accuracies of 19 corruptions on CIFAR-10-C. For brevity, DA-PGD, DA-ZOO-Input, DA-PL and DA-Direct are abbreviated as D-PG, D-Z-I, D-PL and D-Di respectively.

C	Deployed	DINE	BETA	D-PG	D-Z-I	D-PL	D-Di	SODA	SODA-R	MA-SO
GN	51.28	56.86	62.85	28.34	56.94	52.18	48.80	72.86	85.73	84.18
ShN	56.02	58.44	64.75	29.52	53.53	56.63	54.59	74.47	86.07	85.22
IN	42.98	47.25	53.36	22.23	38.02	44.38	41.12	56.65	83.88	75.30
SpN	57.15	59.41	65.61	27.36	57.33	57.07	56.04	73.48	85.58	84.52
DB	88.16	88.14	86.94	16.39	85.91	88.67	85.94	90.95	91.46	90.92
GlB	49.21	53.31	58.38	17.48	43.70	49.75	44.68	66.29	76.99	76.48
MB	76.62	77.25	79.27	17.76	63.41	77.35	75.16	86.69	90.98	87.39
ZB	89.14	89.37	88.86	17.76	81.38	89.73	87.52	91.23	92.50	92.56
GaB	84.59	84.66	84.65	15.87	76.32	85.74	84.38	91.71	93.09	90.98
SW	78.06	78.03	77.42	36.16	77.44	78.62	75.53	83.85	89.00	86.00
FR	71.75	72.39	72.96	23.15	72.21	72.24	70.41	82.98	87.46	87.64
FG	70.58	71.84	73.60	11.56	56.65	71.56	71.71	83.06	84.49	82.78
BR	92.98	92.85	91.28	41.57	89.80	92.75	90.18	92.91	93.07	92.17
CT	86.72	86.74	84.64	15.06	87.92	87.95	87.15	92.48	93.73	91.80
ET	76.64	77.35	78.02	18.32	67.44	77.04	71.99	79.75	82.93	81.98
PX	52.12	58.46	64.50	27.95	58.65	52.35	49.55	87.24	90.23	89.18
JC	80.55	80.93	80.70	29.20	80.19	81.03	78.60	86.13	87.93	87.05
SP	77.66	77.11	77.80	30.20	75.29	77.86	75.69	82.66	88.91	85.61
SA	93.13	92.90	92.98	42.09	92.16	92.68	90.02	92.94	93.58	92.43

Table 14: Accuracies of 19 corruptions on CIFAR-100-C. For brevity, DA-PGD, DA-ZOO-Input, DA-PL and DA-Direct are abbreviated as D-PG, D-Z-I, D-PL and D-Di respectively.

C	Deployed	DINE	BETA	D-PG	D-Z-I	D-PL	D-Di	SODA	SODA-R	MA-SO
GN	19.21	20.17	20.89	5.31	9.12	19.13	16.73	41.01	53.78	57.12
ShN	22.13	23.02	24.23	5.28	11.17	21.87	19.49	42.46	55.09	58.38
IN	12.26	11.50	11.78	3.70	11.76	12.38	10.32	20.70	49.19	47.81
SpN	23.37	23.84	25.02	4.69	13.81	23.27	20.39	40.33	53.05	58.40
DB	60.39	57.75	56.31	3.24	33.07	60.00	55.37	67.05	67.84	68.86
GlB	17.74	17.81	18.58	3.01	9.00	17.22	12.90	29.90	42.03	49.79
MB	45.79	43.98	43.20	3.61	29.43	46.38	43.43	59.24	67.13	63.93
ZB	61.64	59.07	57.06	3.42	34.66	61.98	57.18	65.15	66.20	70.38
GaB	54.40	51.94	49.67	3.20	30.22	55.09	51.40	68.68	70.05	69.40
SW	45.47	44.82	43.18	6.08	39.08	44.88	40.18	50.60	58.47	58.22
FR	39.77	39.65	39.52	3.49	39.62	39.77	37.11	52.21	57.81	60.88
FG	31.94	31.23	30.71	1.43	8.93	31.66	31.07	48.49	55.59	54.49
BR	71.18	69.67	65.47	6.53	55.12	70.23	64.35	70.23	70.75	70.12
CT	49.10	46.10	43.35	1.23	21.50	51.56	48.43	72.18	73.27	69.86
ET	40.45	39.86	38.89	3.29	32.48	39.66	32.71	40.16	49.54	56.44
PX	27.77	27.87	29.36	4.24	17.36	27.72	24.85	55.65	62.41	66.24
JC	49.98	49.50	48.39	5.91	42.32	50.36	45.79	56.25	59.99	63.66
SP	44.18	43.80	42.47	5.03	31.65	44.56	39.20	50.70	61.74	61.93
SA	69.97	68.25	64.69	6.24	62.99	69.64	64.36	69.56	71.03	71.59

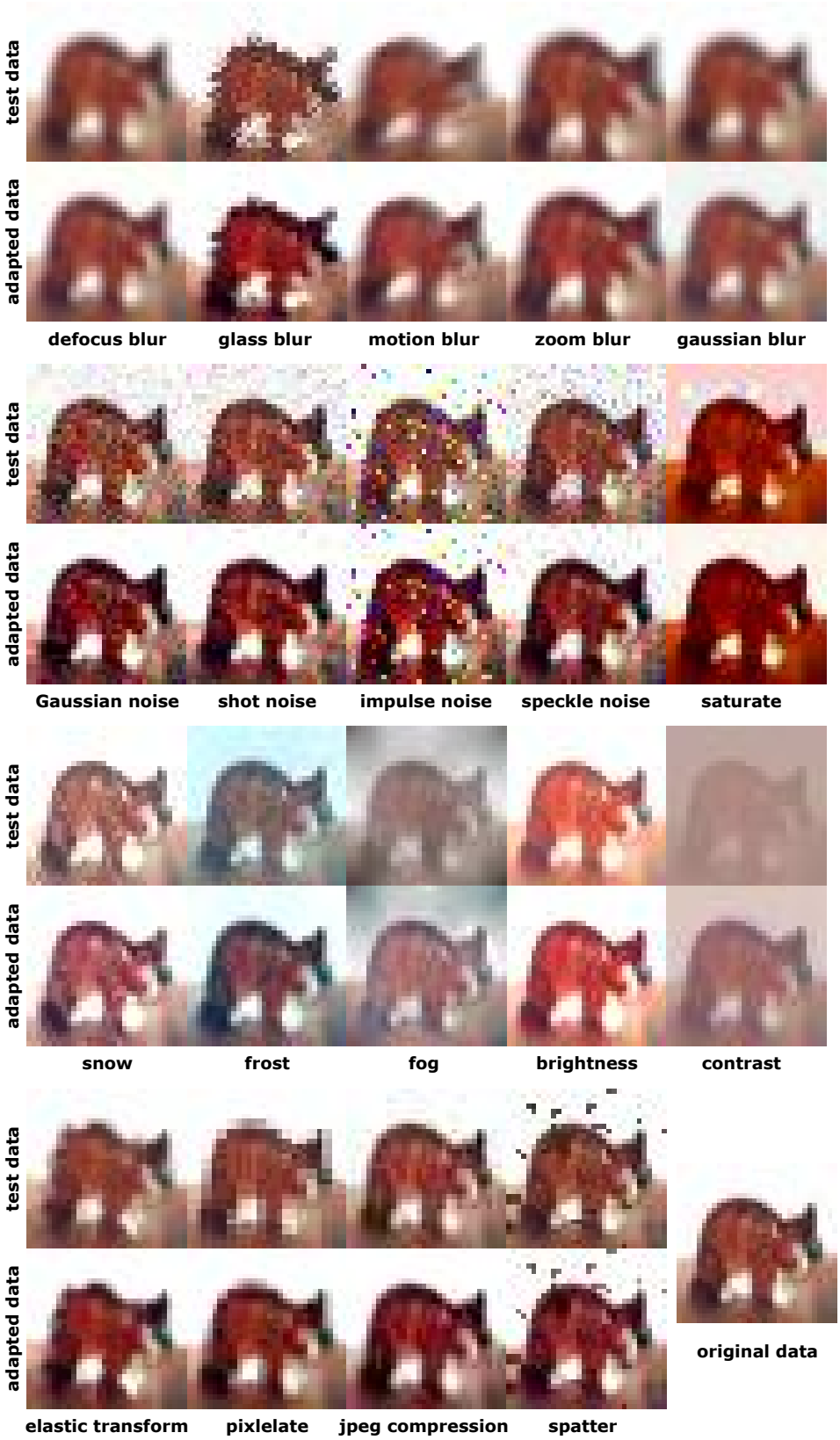


Figure 9: Examples of test data and adapted data using SODA for 19 corruptions in CIFAR-10-C. The bottom-right data is the original data in CIFAR-10 test dataset without corruption.