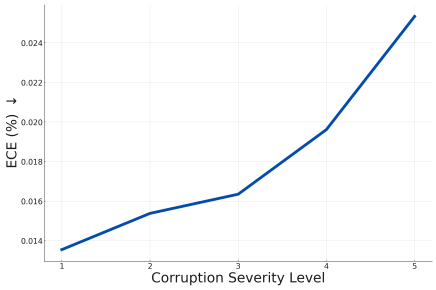
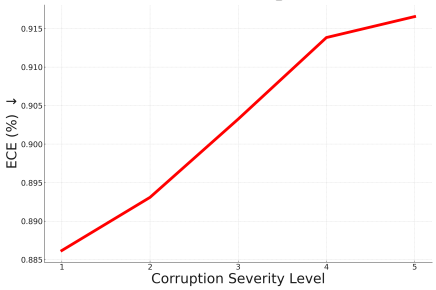
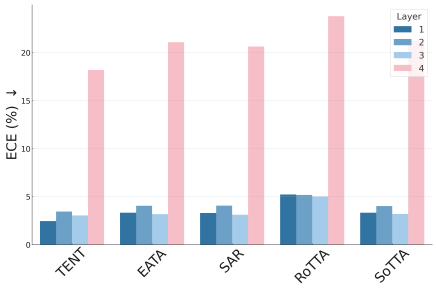
Supplementary Materials
Additional Studies



(a) Trend of ECE across Dist. shift severity. (Correct Samples)



(b) Trend of ECE across Dist. shift severity. (Incorrect Samples)



(c) Ablation for feature extraction layer.

Figure 7: Comparison of calibration errors across various corruption severity levels for both correct (7a) and incorrect (7b) samples. / Comparison of calibration errors across feature extraction layer for generating style variants of SICL (7c)

.

Observation - Impact of the distribution shifts on TTA We illustrate the impact of distribution shift intensity on calibration performance under TTA environment with representative TTA algorithm TENT (Wang et al. 2021) by plotting the calibration error across corruption severities of all 15 corruptions in the CIFAR10-C dataset in the Figure 7. Here, we calculated the ECE for correctly predicted samples and incorrectly predicted samples separately, and plotted them in Figure 7a and Figure 7b, respectively since we wanted to examine the trends of calibration error independently of the model's accuracy. Regardless of prediction correctness, calibration error rises consistently with increasing shift severity, as models tend to misinterpret novel samples as previously learned representations when shifts deviate further from the training distribution.

Ablation Study - Impact of the feature extraction layer We present the ECE for each TTA method in the Figure 7c, based on the feature extraction layer used to extract style statistics when creating SICL's Style-shifted Variants. The experiments were conducted on all 15 corruptions of CIFAR-10C, and the average ECE was recorded. Each layer corresponds to a ResNet block. As shown in the figure, extracting features from layers 1, 2, and 3 results in minimal differences in ECE. However, extracting features from layer 4 to create style variants leads to significantly higher errors. This can be attributed to the fact that deeper layers encode more content information (Pan et al. 2018), rather than style. Consistent with the observations in Section 5.3 of the main paper, this result further demonstrates that when the content of an instance is distorted and used in the ensemble of candidate representations, the method fails to perform proper instance-wise uncertainty estimation.

Further Discussions

Comparison with previous methods

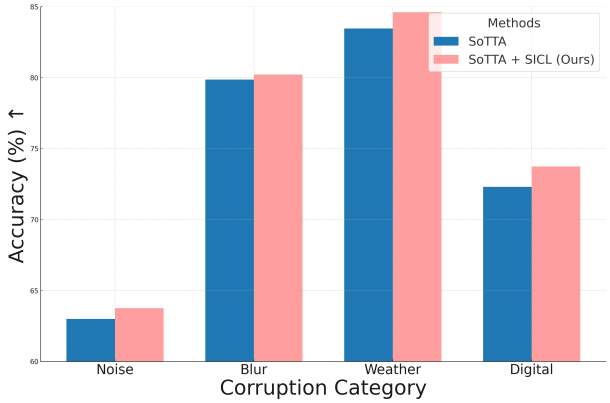
Temperature Scaling-based Methods Temperature scaling-based calibration methods (Guo et al. 2017; Gal and Ghahramani 2016a; Wang et al. 2020; Hu et al.; Park et al. 2020) assume a fixed model and distribution, scaling the entire target dataset with a single temperature. As a result, they fail to effectively address varying characteristics of each test instances under diverse distribution shifts and changing model during TTA. Moreover, obtaining the temperature often requires additional training with labeled source data (Guo et al. 2017; Wang et al. 2020; Park et al. 2020), or pseudo-data (Hu et al.), which are unavailable at test time and adds a burden on top of the adaptation process. However, unlike these methods, SICL computes prediction confidence on an instance-by-instance basis, enabling effective uncertainty estimation in highly variable test streams. Since it does not rely on trainable temperatures, it eliminates the need for memory-intensive backward computations or additional trainable parameters.

Ensemble-based Methods Ensemble-based calibration methods (Gal and Ghahramani 2016a; Lakshminarayanan, Pritzel, and Blundell 2017) which aggregate stochastic predictions of candidate representations, enable instance-wise calibration, making them capable of handling various non-fixed distributions. However, MC Dropout (Gal and Ghahramani 2016a) can reduce accuracy while averaging logits during dropout inference. And the process of randomly disabling neurons of the model result in representations that lose critical content information (analyzed in Section 5.3). Also, Deep Ensembles (Lakshminarayanan, Pritzel, and Blundell 2017) require training several models to ensemble, which is undesirable in the resource-restricted test-time adaptation environments. However, unlike these methods, SICL leverages only the forward pass of an online model to compute the consistency ratio across style variants and derives the final prediction confidence. This post-hoc calibration approach

does not require additional training or external models and does not compromise accuracy. Furthermore, style-shifted variants of SICL serve as excellent candidate representations because they are generated by extracting feature-level style statistics and perturbing them with bounded Gaussian noise. These representations preserve content information while diversifying only the style.

Use case of SICL

(a) OOD Detection performance (AUROC(%))



(b) Impact on classification performance (Accuracy(%))

Figure 8: Comparison of OOD Detection performance with baselines across various corruption categories (8a) under the noisy stream of CIFAR-10C. / Effect of SICL appilied to SoTTA under the noisy stream of CIFAR-10C (8b)

.

Out-of-distribution Detection SICL can naturally be extended to Out-of-Distribution (OOD) detection during TTA. It has been observed in prior study (Gong et al. 2023), OOD samples injected during TTA can lead to performance degradation and even model failures. SICL, allowing models to produce reliable predictions, can help mitigate such shortfalls, and give model vendors the freedom of choosing diverse TTA methodologies - instead of relying on a single strategy specifically designed for such scenarios such as SoTTA (Gong et al. 2023). In the following paragraphs, we (1) highlight the high-level working of SICL for OOD detection and empirically validate the assertion via (2) OOD detection performance and (3) integration with SoTTA under noisy test streams.

The core concept of SICL lies in estimating correctness likelihood by measuring prediction consistency across style-transformed variants. In-Distribution (ID) samples tend to maintain high consistency than OOD samples, for OOD samples deviate far from the learned content distribution, exhibiting inconsistent predictions under style variants of SICL. Thus, SICL can effectively differentiate between ID and OOD samples via simple thresholding.

We empirically validate this approach by comparing the out-of-distribution detection performance of various calibration methods on the CIFAR-10C *noisy* stream using TENT (Wang et al. 2021), a representative TTA method, based on AUROC. Figure 8a shows that SICL achieved the highest AUROC across all corruption categories.

We further demonstrate the capability of SICL-based OOD detection method in enhancing performance of TTA methods under noisy test streams in Figure 8b. Specifically, we substitute OOD sample filtering process of HUS with SICL - instead of naive filtering of OOD samples based on model's prediction confidence, we apply SICL to calibrate its predictions beforehand. On average, we see accuracy improvements from 75.4% to 76.4% in CIFAR10-C corrupted with noisy samples obtained from MNIST. The result demonstrate that through SICL, the model is prevented from learning incorrect representations from noisy samples, thereby enhancing its prediction accuracy.

Uncertainty Modeling in Object Detection SICL can potentially be applied to object detection models as well, to determine regions where predictions are uncertain. It is known that object detection models such as YOLO (Redmon 2016) are known to exhibit overconfidence issues (Melotti et al. 2022, 2023), similar to our observation in TTA methods, hindering its deployment to risk-sensitive domains. Similar to MC Dropblock (Deepshikha et al. 2021) which utilizes stochastic predictions to model uncertainty in object detection, SICL leverages style perturbations to evaluate prediction consistency, making it possible to identify OOD regions or masks.

Experiment Details

All our experiments were conducted on NVIDIA TITAN RTX and NVIDIA GeForce RTX 3090 GPUs. We use source pre-trained network as an initial model for all datasets. The source models are trained on clean training data to produce the source models. The training process of the source model employs SGD with a momentum of 0.9 and a cosine annealing learning rate scheduler with the initial learning rate 0.1 over 200 epochs, following SoTTA (Gong et al. 2023).

Evaluation Details

To carry out predictive uncertainty calibration in TTA, the model's predictions are calibrated for each online batch after model updates, ensuring that predictions remain reliable as the model adapts to new distributions. For each test batch at time step t, the predicted confidence $\text{conf}^{(t)}$ and accuracy $\text{acc}^{(t)}$ are recorded and accumulated from the beginning to

the current time, providing a basis for cumulative calibration error. Here, we define Cumulative Expected Calibration Error (ECE) specifically for TTA scenarios, extending the standard ECE definition (Naeini, Cooper, and Hauskrecht 2015) to quantify the discrepancy between confidence and accuracy over time by aggregating online model's predictions at each time step. The Cumulative ECE can be expressed as:

$$\frac{1}{n_T}\sum_{t=1}^{n_T}\sum_{k=1}^{K}\frac{|B_k^{(t)}|}{N^{(t)}}\left|\text{acc}_k^{(t)} - \text{conf}_k^{(t)}\right|, \tag{6}$$

where n_T represents the total number of test batches, K is the number of confidence bins, $B_k^{(t)}$ is the set of samples in bin k during the t-th batch, $\text{acc}_k^{(t)}$ and $\text{conf}_k^{(t)}$ are the accuracy and average confidence of $B_k^{(t)}$ and $N^{(t)}$ is the total number of samples in the t-th batch. This formulation allows continuous monitoring of calibration quality over time, supporting TTA scenarios where high confidence alignment with accuracy is critical. Note that all the calibration errors reported in this paper are cumulative ECE.

Dataset Details

CIFAR10-C, CIFAR100-C CIFAR10-C and CIFAR100-C are widely recognized benchmark datasets for evaluating model robustness against various types of corruption. Both datasets include 50,000 training samples and 10,000 test samples, divided into 10 and 100 object classes. To test robustness, both datasets introduce 15 corruption types under 4 categories: Noise, Weather, Digital, and Blur. The corruption types are Gaussian Noise, Shot Noise, Impulse Noise, Brightness, Snow, Frost, Fog, Contrast, Elastic Transformation, Pixelation, JPEG Compression, Defocus Blur, Glass Blur, Motion Blur, and Zoom Blur. We use the highest corruption severity level, level 5 for all our experiments, following prior works (Wang et al. 2021; Gong et al. 2023; Yuan, Xie, and Li 2023a; Schneider et al. 2020).

ImageNet-C ImageNet-C represents another benchmark widely used to assess model resilience when faced with various corruptions, as noted in several studies [1, 27, 36, 38, 39]. The original ImageNet collection (Deng et al. 2009) encompasses 1,281,167 samples for training and 50,000 for testing purposes. Following a similar approach to CIFAR10-C, this dataset applies an identical set of 15 corruption types, generating a total of 750,000 corrupted test images. In our research, we employ the most severe corruption level (level 5), consistent with our CIFAR10-C and CIFAR100-C setup.

Test Scenario Details

- **Benign**: In the *benign* scenario, we followed the default settings adopted by other TTA works. The test stream was constructed as an i.i.d. distribution from a single corruption type (*e.g.* Gaussian noise) without any noisy samples, and the final results were reported as the average performance across all corruption types.
- **Dynamic**: To evaluate TTA calibration in realistic scenarios, we introduce a **dynamic scenario**. Our novel setting shares similarities with prior continual TTA setting, in that it assumes multiple corruption types during test, but differs in the sense that corruption types do not arrive sequentially. Instead, corruption types have temporal correlation generated for each corruption type using a Dirichlet distribution. Figure 9 demonstrates a visualization of our dynamic test stream scenario.

Baseline Details

TTA methods Here, we outline the hyperparameter selection of TTA methods. We adopted the hyperparameters documnented in respective papers or source code repositories.

- **TENT.** We set the learning rate as $LR = 0.001$ for CIFAR-10-C and CIFAR-100-C, while $LR = 0.00025$ for ImageNet-C, adhering to the selection of the original paper. We utilized the original code provided by the authors for implementation.
- **SAR.** We set the learning rate as $LR = (0.00025, 0.001)$ for ResNet and ViT models, repectively. Also, we set the sharpness threshold $\rho = 0.5$, and entropy threshold $E_0 = 0.4 \times \ln|\mathcal{Y}|$, where $|\mathcal{Y}|$ is the number of classes. We utilized the original code provided by the authors for implementation.
- **EATA.** We set the learning rate as $LR = (0.005, 0.005, 0.00025)$, cosine similaritty threshold $\epsilon = (0.4, 0.4, 0.05)$, tradeoff parameter $\beta = (1, 1, 2000)$ for CIFAR-10-C, CIFAR-100-C and ImageNet-C, respectively. We set the entropy constant $E_0 = 0.4 \times \ln|\mathcal{Y}|$, where $|\mathcal{Y}|$ is the number of classes. We take 2000 samples for calculating Fisher importance, and referred to the official code for implementation.
- **RoTTA.** We set the learning rate as $LR = 0.001$ and $\beta = 0.9$, and BN-statistics update moving average update rate $\alpha = 0.05$, and Teacher model's exponential moving average updating rate as $\nu = 0.001$, and timeliness parameter $\lambda_t = 1.0$, and uncertainty parameter $\lambda_u = 1.0$, following the authors' selections.
- **SoTTA.** We used a BN momentum of $m = 0.2$, and learning rate of $LR = 0.001$ with a single adaptation epoch. We set the HUS size to 64 and the confidence threshold $C_0 = (0.99, 0.66, 0.33)$ for CIFAR10-C, CIFAR100-C, and ImageNet-C, respectively. We set entropy-sharpness L_2-norm constraint $\rho = 0.5$ following the suggestion.

Calibration methods For all calibration baselines and SICL(ours), we performed a hyperparameter search upon representative corruption (Gaussian Noise) and fixed its value across all corruption types.

- **Vanilla TS.** Using the validation set, we optimize the temperature through scikit-learn's optimizer, with initial temperature set as 2.0.
- **MC Dropout.** We set a fixed dropout ratio of 0.3 and the number of inference steps $N = 20$ for all our experiments.
- **TransCal.** As in vanilla TS, we optimize the temperature with the Sequential Least Squares Quadratic Programming (SLSQP) optimization method, adopting the implementation in PseudoCal.

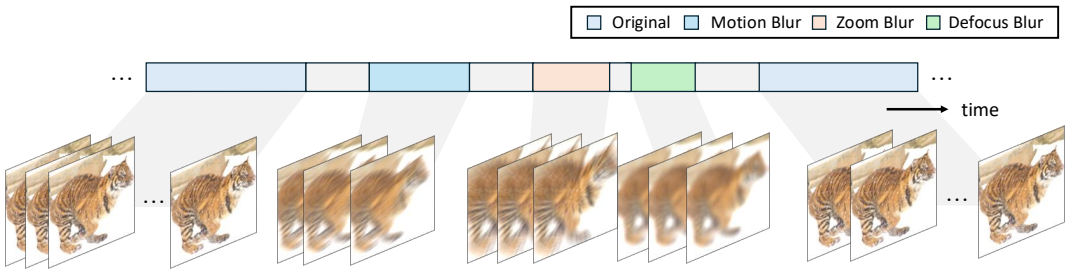


Figure 9: An illustration of the *dynamic* test stream over the time axis that could occur when filming moving objects.

- **PseudoCal.** We set the learning rate of temperature as 0.05, an optimal choice from the CIFAR-10C dataset. We utilized the original code provided by the authors for implementation.
- **SICL(Ours).** We used a fixed number of style variants $N = 20$ and the sensitivity hyperparameter $n = 3$ for all our experiments.

Style Shifting methods

- **MixStyle.** For the qualitative analysis and ablation studies, we set an alpha hyperparameter from beta distribution for deciding mixing ratio as 0.1, adopting the original implementation in MixStyle (Zhou et al. 2021).

Additional Results

Additionally, to evaluate the performance on a lightweight model, we conducted additional experiments on CIFAR-10C and CIFAR-100C using ResNet-18 as the backbone network. Tables 5 and 6 present the results for CIFAR-10C and CIFAR-100C, respectively. The test stream was evaluated under the Benign setup, while all other settings remained the same.

Table 3: Expected Calibration Error (ECE) (%) of uncertainty estimation on CIFAR-10C dataset of all corruption types under benign stream using various TTA methods. **Bold** numbers represent the lowest error.

TTA Method	Baseline	Noise			Blur				Weather				Digital				Avg. (↓)
		Gau.	Shot	Imp.	Def.	Gla.	Mot.	Zoom	Snow	Fro.	Fog	Brit.	Cont.	Elas.	Pix.	JPEG	Avg.
TENT (Wang et al. 2021)	Vanilla TS	25.04	23.44	31.49	12.11	31.97	12.80	11.16	16.52	17.36	12.25	9.03	11.90	21.98	16.99	24.30	20.15
	MC Dropout	15.21	14.45	19.86	7.55	22.62	7.84	6.50	10.50	11.18	7.24	5.26	8.36	13.61	10.26	16.44	11.79
	TransCal	17.32	18.64	18.12	21.62	11.42	15.73	19.23	6.90	12.02	12.58	5.21	34.55	4.93	15.03	4.23	13.85
	PseudoCal	8.78	7.88	12.62	3.40	14.66	3.41	3.61	4.36	4.74	3.98	3.97	4.50	6.27	3.73	8.58	6.30
	SICL(Ours)	4.78	5.23	7.55	1.20	3.01	1.52	1.51	1.30	1.41	2.17	0.98	4.40	3.52	3.05	1.69	**2.89**
EATA (Niu et al. 2022)	Vanilla TS	33.13	30.41	39.24	11.68	36.09	13.42	12.53	18.60	19.73	14.74	9.16	12.87	23.18	21.43	28.06	21.55
	MC Dropout	15.00	14.27	18.99	5.48	17.99	6.28	5.78	8.61	9.02	6.05	3.91	5.18	11.13	9.23	14.03	10.06
	TransCal	17.12	17.09	15.60	21.95	11.22	15.12	17.78	5.55	9.53	9.97	5.15	34.12	4.01	13.67	4.04	12.86
	PseudoCal	8.43	6.78	11.64	4.34	10.45	3.39	3.62	3.16	2.36	3.41	4.21	3.79	4.25	2.91	6.23	5.26
	SICL(Ours)	7.80	8.77	6.24	1.90	3.66	1.90	1.30	1.15	3.08	2.45	1.11	2.08	4.03	4.13	3.42	**3.54**
SAR (Niu et al. 2023a)	Vanilla TS	31.85	30.07	35.88	11.63	33.48	13.42	12.87	18.54	19.97	14.82	9.19	12.90	23.26	21.48	28.15	21.15
	MC Dropout	15.30	13.97	16.88	5.46	17.82	6.21	5.77	8.66	8.93	6.15	3.88	5.30	11.24	9.27	14.26	9.94
	TransCal	17.11	17.13	16.59	21.99	11.19	15.13	17.76	5.67	9.41	9.90	5.12	34.06	4.02	13.66	4.08	13.52
	PseudoCal	7.79	6.69	10.73	4.21	10.19	3.20	3.68	3.36	2.59	3.38	4.20	3.64	4.10	2.75	6.32	5.12
	SICL(Ours)	7.60	8.77	6.35	1.94	3.26	1.93	1.32	1.21	3.01	2.43	1.11	2.22	3.93	4.34	3.42	**3.52**
RoTTA (Yuan, Xie, and Li 2023a)	Vanilla TS	33.02	30.41	39.16	11.75	34.57	13.06	12.00	18.46	19.85	14.19	8.77	15.34	22.86	21.44	27.53	21.50
	MC Dropout	13.22	12.26	17.38	5.68	16.67	5.10	4.79	7.59	7.68	4.71	2.96	10.14	9.51	9.09	13.04	9.32
	TransCal	18.15	17.84	15.19	21.35	11.28	14.57	17.83	5.33	9.05	10.28	5.12	22.90	3.60	13.68	4.26	12.70
	PseudoCal	8.50	7.71	12.46	4.42	10.47	3.45	4.30	3.48	3.11	3.65	4.41	4.17	4.31	3.49	6.47	5.63
	SICL(Ours)	8.88	9.20	5.79	2.39	3.57	2.82	2.14	1.49	2.65	4.21	1.79	1.35	4.24	3.91	3.57	**4.68**
SoTTA (Gong et al. 2023)	Vanilla TS	33.46	30.95	39.32	11.63	35.08	13.42	12.87	18.54	19.97	14.82	9.19	12.90	23.26	21.48	28.15	21.66
	MC Dropout	15.84	14.44	19.48	5.30	19.08	6.32	9.27	5.89	9.26	7.04	4.03	5.45	11.65	10.67	14.81	10.57
	TransCal	17.05	17.18	15.43	21.99	11.23	15.13	17.76	5.67	9.41	9.90	5.12	34.06	4.02	13.66	4.09	13.44
	PseudoCal	8.91	11.94	6.86	4.21	10.38	3.28	3.61	3.32	2.68	3.39	4.01	3.64	4.31	2.82	6.37	5.35
	SICL(Ours)	8.20	6.16	8.77	1.94	3.67	1.93	1.32	1.21	3.01	2.43	1.11	2.22	3.93	4.34	3.42	**3.58**

Table 4: Expected Calibration Error (ECE) (%) of uncertainty estimation on CIFAR-100C dataset of all corruption types under benign stream using various TTA methods. **Bold** numbers represent the lowest error.

TTA Method	Baseline	Noise			Blur				Weather				Digital				Avg. (↓)
		Gau.	Shot	Imp.	Def.	Gla.	Mot.	Zoom	Snow	Fro.	Fog	Brit.	Cont.	Elas.	Pix.	JPEG	Avg.
TENT (Wang et al. 2021)	Vanilla TS	49.10	33.16	38.83	21.82	35.29	23.21	20.77	27.41	27.68	24.84	19.84	23.03	28.07	24.58	30.82	28.56
	MC Dropout	5.85	6.20	4.31	5.48	6.91	6.08	6.11	6.06	5.14	5.26	7.26	6.30	5.33	7.05	4.76	5.79
	TransCal	34.10	35.68	32.95	49.91	34.78	44.14	49.99	39.13	43.38	42.19	36.81	48.71	34.12	48.83	34.72	40.63
	PseudoCal	8.78	8.76	7.80	12.83	8.93	11.80	13.77	11.12	11.49	12.17	13.12	11.92	11.24	12.85	10.16	11.12
	SICL(Ours)	6.65	6.88	7.12	2.84	2.50	5.22	7.07	7.04	5.20	3.31	6.27	4.86	8.34	7.36	2.31	**5.49**
EATA (Niu et al. 2022)	Vanilla TS	52.76	50.97	55.07	31.66	53.62	38.65	32.06	43.69	38.88	37.74	29.38	30.65	46.45	40.09	48.30	41.99
	MC Dropout	15.49	16.45	14.46	17.08	17.37	20.27	15.91	19.56	17.05	18.38	25.60	16.96	16.53	15.85	17.17	17.64
	TransCal	18.91	21.93	12.13	34.65	11.87	28.97	29.74	16.51	27.08	26.56	21.98	36.11	11.80	24.66	13.14	22.40
	PseudoCal	10.74	10.22	10.26	9.69	13.61	9.61	9.47	9.42	8.70	8.89	9.74	8.73	9.87	8.88	9.64	9.83
	SICL(Ours)	5.83	7.55	12.98	2.80	17.20	3.79	4.82	5.19	8.84	5.35	3.84	3.79	11.19	4.12	12.95	**7.61**
SAR (Niu et al. 2023a)	Vanilla TS	33.74	32.19	37.25	22.01	35.69	24.16	21.87	28.07	28.59	25.84	20.05	20.50	29.07	24.77	31.38	27.68
	MC Dropout	8.62	9.40	7.40	10.59	11.29	11.87	8.55	11.14	9.93	11.87	11.48	24.36	11.15	10.39	10.89	11.24
	TransCal	34.93	36.84	35.36	51.23	36.73	44.58	49.88	39.86	43.05	42.57	37.82	51.26	34.71	49.84	35.86	41.57
	PseudoCal	7.94	7.62	7.48	9.98	7.19	8.49	10.04	9.35	8.49	9.48	10.25	10.43	9.60	9.59	8.36	8.95
	SICL(Ours)	7.49	8.07	6.22	6.76	2.74	3.15	4.51	2.07	4.98	6.98	4.28	6.11	8.69	8.06	2.59	**5.64**
RoTTA (Yuan, Xie, and Li 2023a)	Vanilla TS	39.81	38.98	42.62	22.64	38.83	24.21	22.91	30.58	30.94	27.92	20.83	23.60	30.03	28.49	35.62	30.53
	MC Dropout	11.14	12.28	9.32	7.65	9.28	6.91	9.10	7.64	7.36	11.03	7.71	9.92	10.82	10.88	7.86	8.59
	TransCal	25.95	25.82	22.19	46.74	27.02	40.66	45.13	31.15	29.04	36.70	32.89	17.73	29.29	39.19	24.37	31.62
	PseudoCal	7.98	8.64	8.04	9.71	8.42	9.29	9.86	10.15	9.01	10.46	10.39	9.25	9.83	10.08	8.47	9.30
	SICL(Ours)	9.07	10.08	6.84	7.70	6.05	6.46	5.95	5.35	5.31	8.89	4.16	3.78	9.05	10.52	3.10	**8.37**
SoTTA (Gong et al. 2023)	Vanilla TS	40.92	39.41	43.62	23.11	39.04	25.67	24.05	31.60	32.20	28.78	21.65	23.69	31.68	29.13	36.69	31.25
	MC Dropout	8.73	9.87	8.57	5.77	6.40	6.49	5.67	6.71	5.82	6.08	5.67	9.46	6.93	7.08	6.04	6.57
	TransCal	27.28	28.11	20.47	48.62	29.20	40.79	46.41	33.64	37.19	36.68	35.43	45.72	30.28	42.49	25.45	35.50
	PseudoCal	7.44	10.58	7.99	9.07	9.79	9.12	9.05	9.46	9.37	9.08	10.06	9.46	9.37	9.80	9.25	9.03
	SICL(Ours)	7.65	9.07	5.97	5.77	4.89	4.53	5.35	5.08	6.13	7.18	4.95	4.10	7.71	9.25	3.37	**6.19**

License of Assets

Datasets

CIFAR10/CIFAR100 (MIT License), CIFAR10-C/CIFAR100-C (Creative Commons Attribution 4.0 International), MNIST (CC-BY-NC-SA 3.0), ImageNet-C (Apache 2.0).

Codes

Torchvision for ResNet-18, ResNet-50, and ResNet-101 (Apache 2.0), the vit-pytorch repository from lucidrains (MIT License), the official repository of CoTTA (MIT License), the official repository of TENT (MIT License), the official repository of EATA (MIT License), the official repository of SAR (BSD 3-Clause License), the official repository of RoTTA (MIT License), the official repository of SoTTA (MIT License), and the official repository of PseudoCal (MIT License).