
A SUPPLEMENTARY

A.1 DETAILS ON INTRANSIGENT TEACHER EXPERIMENT FROM SECTION 3

Our preliminary intransigent teacher experiment is conducted on ImageNet-C, a long ImageNet-C scenario (20x loop of standard ImageNet-C testing sequence), and CCC benchmarks. We utilize the same model as for our main experiments on ImageNet-based benchmarks - ResNet50 with pre-trained weights from the *RobustBench* (Croce et al., 2021) model zoo. We use a single loss within the teacher-student framework for the model adaptation during test time - either consistency loss from CoTTA (Consistency) or contrastive loss from AdaContrast (Contrastive). Any other components of the mentioned state-of-the-art methods are not included. Batch normalization statistics are recalculated for each batch. For both of the tested approaches, we use the SGD optimizer with a learning rate of 0.00025.

A.2 BASELINES IMPLEMENTATION DETAILS

The experiments were conducted using the code repository of the previous test-time adaptation works (Marsden et al., 2024; Döbler et al., 2022). It provides the implementation of every tested state-of-the-art method. In terms of hyperparameters, we followed the implementations for tests on the typical batch size of 64.

TENT (Wang et al., 2021), EATA (Niu et al., 2022), SAR (Niu et al., 2023), and RDUMB (Press et al., 2023) use Adam optimizer with a learning rate of 0.001 for CIFAR10-C and SGD optimizer with a learning rate of 0.00025 for other benchmarks. AdaContrast (Chen et al., 2022) utilizes an SGD optimizer with a learning rate set to 0.0002 for all of the benchmarks. CoTTA (Wang et al., 2022) uses Adam optimizer with a learning rate of 0.001 for CIFAR10-C and SGD with a learning rate of 0.01 for the rest of the benchmarks. Adam optimizer with a learning rate set to 0.001 is used by RoTTA (Yuan et al., 2023a) for all of the tested datasets. MEMO (Zhang et al., 2022) uses an SGD optimizer with a learning rate of 0.005 for CIFAR10-C and 0.00025 for other datasets. PETAL (Brahma & Rai, 2023) in the original paper uses Adam optimizer with a learning rate of 0.001 for CIFAR10-C and SGD with a learning rate of 0.01 for other datasets. However, since we often experienced poor performance using these values on long scenarios, we utilized 10 times lower learning rates.

The learning rate used in experiments with batch size set to 10 was adjusted accordingly by scaling it linearly.

CoTTA (Wang et al., 2022) and PETAL (Brahma & Rai, 2023) methods update the student network using a consistency loss between the student and teacher. If the prediction confidence of the source model is below a certain threshold, the teacher’s predictions are averaged over 32 different augmentations of the image which adds 31 additional forward operations of the neural network for each batch. It creates a significant computation overhead and causes the methods to be significantly slower, compared to other state-of-the-art methods. It is especially problematic for long adaptation sequence scenarios, which were the main part of our experiments. Our tests indicate that using a single augmentation does not alter the results notably. Therefore, for the ease of experimentation, we reduce the number of augmentations to 1.

The learning rate selection process for Figure 6 (right) was conducted using the Oracle method.

A.3 DETAILS ON MEMO RESULTS ON CCC BENCHMARK FROM TABLE 3

The result is based on the first 623,000 images of the benchmark, providing an initial estimate of the method’s accuracy. However, due to the benchmark’s extensive size (7,500,000 images) and the method’s requirement for a batch size of 1, we were unable to complete the full experiment in time. We estimate that processing the entire dataset will require approximately 972 hours on a single NVIDIA GeForce RTX 4080 GPU. This substantial time requirement underscores the method’s significant computational inefficiency.

A.4 COMPUTE DETAILS

All experiments were conducted on a single GPU. We utilized either NVIDIA A100 with 40GB of memory or NVIDIA GeForce RTX 4080 with 16GB of memory. Execution time of experiment greatly varied and was dependent on the dataset, scenario (standard or long), tested method and batch size. The fastest experiments took about 30 minutes, whereas the longest lasted up to 36 hours.

A.5 DISCUSSION ON CoTTA AND I-CoTTA PERFORMANCE ON IMAGENET-C (L) AND IMAGENET-R (L)

I-CoTTA underperforms compared to the original CoTTA on ImageNet-C (L) and ImageNet-R (L) with a batch size of 64 and architectures with batch normalization layers, as shown in Table 3 and Table 4. The accuracy drops by 17.4 and 10.8 percentage points, respectively. We attribute this to CoTTA’s exceptional performance in these specific scenarios, where it outperforms all other tested methods and achieves a stable performance improvement as presented in Figure 4 (left). The additional regularization from IT doesn’t enhance stability in this case. Instead, it over-regularizes the student model, hindering its adaptation capability. This case, while unusual for CoTTA (considering other CoTTA results), demonstrates that IT isn’t universally effective. However, it’s crucial to note that even in this case, IT still outperforms the source model. Our focus is on improving the overall reliability of TTA across all settings, not just in specific scenarios where certain methods may excel. Also, note that COTTA does not perform that well on architectures without batch normalization layers.

A.6 WALL-TIME RESULTS

Table A.1: The wall-clock time (seconds) for processing 10,000 images of CIFAR10C on a single RTX 4080 GPU.

Method	Time [s]
Source	3.4
MEMO	508.4
AdaContrast	25.3
I-AdaContrast	25.0
CoTTA	40.7
I-CoTTA	40.2
RoTTA	27.7
I-RoTTA	27.5

A.7 RESULTS WITH DIFFERENT ARCHITECTURES AND LEARNING RATES

Table A.2 presents additional results using different neural network architectures. The learning rate was tuned by the Oracle method to provide favorable conditions for the original TTA approaches and ensure they work correctly. **All results from the learning rate selection process are in Table A.3.** The intransigent teacher is able to improve the test-time adaptation accuracy on long sequences for all of the compared models even when the original methods have tuned learning rates specifically for tested sequence length.

Table A.2: Classification accuracy [%] for long scenarios on CIFAR10-C and ImageNet-C with different neural network architectures. The value in superscript indicates the improvements over the baseline. The learning rate parameter is adjusted using the Oracle method. **The batch size is equal to 64.**

	CIFAR10-C (L)		ImageNet-C (L)		
	ResNet26GN	ResNeXt-50	ViT-B16	SwinViT-T	ConvNeXt tiny
Source	67.3	21.1	39.8	28.3	29.1
AdaContrast (Chen et al., 2022)	75.7	38.8	41.5	30.6	33.4
I-AdaContrast	79.6 ^{+3.9}	42.7 ^{+3.9}	43.5 ^{+2.0}	30.9 ^{+0.3}	32.5 ^{-0.9}
CoTTA (Wang et al., 2022)	57.0	42.1	41.7	28.4	29.1
I-CoTTA	67.3 ^{+10.3}	38.3 ^{-3.8}	40.7 ^{-1.0}	28.9 ^{+0.5}	30.5 ^{+1.4}
RoTTA (Yuan et al., 2023a)	70.2	35.6	40.6	28.8	29.0
I-RoTTA	72.5 ^{+2.3}	36.2 ^{+0.6}	42.9 ^{+2.3}	28.9 ^{+0.1}	29.7 ^{+0.7}

Table A.3: Classification accuracy [%] for long scenarios on CIFAR10-C and ImageNet-C with different neural network architectures and learning rates with the batch size equal to 64. Intransigent versions are much more robust to changes in hyperparameters.

	LR	CIFAR10-C (L)		ImageNet-C (L)		
		ResNet26GN	ResNeXt-50	ViT-B16	SwinViT-T	ConvNeXt tiny
Source	-	67.3	21.1	39.8	28.3	29.1
AdaContrast	0.001	75.7	20.0	29.6	13.0	17.5
	0.0002	74.7	20.0	32.1	15.1	18.2
	0.00025	75.1	20.3	31.8	14.4	18.2
	3.125e-5	74.3	25.6	39.0	21.9	22.4
	1e-6	72.0	38.8	41.5	29.6	32.0
	1e-7	68.4	33.5	40.7	30.6	33.4
	1e-8	68.1	32.1	40.0	28.7	31.2
I-AdaContrast	0.001	79.6	39.5	42.1	30.4	32.4
	0.0002	79.4	42.7	43.5	30.8	32.5
	0.00025	79.5	42.4	43.4	30.8	32.5
	3.125e-5	77.7	42.3	43.1	30.9	32.3
	1e-6	73.0	37.6	42.0	30.9	31.7
	1e-7	69.2	33.4	41.0	30.3	30.5
	1e-8					
CoTTA	0.01	12.3	57.1	26.2	0.1	0.1
	0.001	16.6	42.1	34.5	26.3	0.2
	0.00025	14.8	39.2	38.7	25.5	19.3
	3.125e-5	26.6	39.3	41.7	28.4	22.2
	1e-6	56.2	33.7	40.0	27.0	29.1
	1e-7	57.0	33.0	39.4	28.0	29.0
	1e-8	57.0	32.9	39.3	28.2	29.1
I-CoTTA	0.01	26.9	38.3	30.9	27.5	16.3
	0.001	61.7	35.9	39.9	28.9	27.6
	0.00025	62.1	36.0	40.1	28.7	29.3
	3.125e-5	62.1	35.7	40.7	28.6	29.8
	1e-6	67.3	33.6	40.4	28.3	30.5
	1e-7	67.3	33.0	39.9	28.3	28.3
	1e-8					
RoTTA	0.001	66.0	16.2	36.2	7.3	18.7
	0.00025	68.5	19.7	34.8	7.9	16.8
	3.125e-5	70.2	35.6	36.5	13.9	16.3
	1e-6	68.5	33.6	40.6	28.8	26.8
	1e-7	67.9	31.2	40.0	28.4	29.0
	1e-8	67.3	30.8	39.8	28.3	29.0
	1e-9					
I-RoTTA	0.001	72.5	35.2	42.9	27.7	29.7
	0.00025	72.4	36.2	42.6	27.3	29.7
	3.125e-5	71.2	34.3	42.1	27.7	29.0
	1e-6	68.9	28.3	40.7	28.9	28.5
	1e-7	67.9	26.1	40.0	28.4	29.0
	1e-8					
	1e-9					

A.8 EFFECTS OF INTRANSIGENCE AMOUNT EXTENDED EXPERIMENT

To signify the point of Section 4.2, Figure A.1 shows results where the test sequence was extended to 100 loops of common CIFAR10-C. It verified that CoTTA with ET and $\beta = 0.9999$ degrades below the performance of IT, given enough samples in the test sequence. This observation highlights a significant issue of TTA methods, as they can face test sequences of arbitrary lengths after deployment.

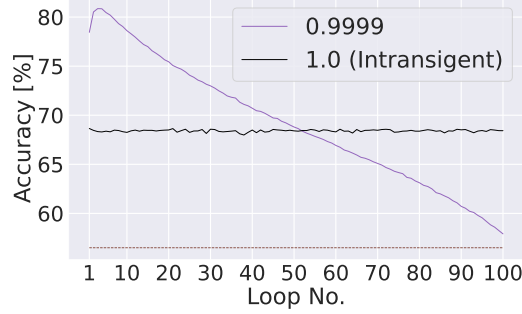


Figure A.1: Mean accuracy [%] of CoTTA with varying β for each loop of common CIFAR10-C testing sequence repeated 100 times. The brown dashed line indicates the accuracy of the source model as a reference.

A.9 TUNING LEARNING RATE VALUE FOR LONG SCENARIOS.

We investigated whether tuning the learning rate, arguably the most crucial hyperparameter, could enhance the performance of baseline methods in long adaptation scenarios. Following a realistic approach, we employed an Oracle technique on ImageNet-C (L) as a reference benchmark (inspired by Rusak et al. (2022), we call it Transfer IN-C) and applied the selected learning rate across all datasets. The results, presented in Table A.4, reveal the complexity of hyperparameter optimization in test-time adaptation.

Our findings shows the challenges of hyperparameter tuning. For instance, CoTTA achieved superior accuracy with its default learning rate compared to the tuned version. While AdaContrast and RoTTA showed improvements with optimized learning rates, our IT approach consistently outperformed these methods, even when they were specifically tuned for long-sequence adaptation. These results underscore both the difficulty of hyperparameter selection and the robust performance of our IT method across varying conditions.

Table A.4: Classification accuracy [%] for long scenarios with the learning rate (LR) parameter tuned. LR value **Default** means that the default LR value for the method was used. **Transfer IN-C** indicates that the LR is tuned utilizing the ImageNet-C benchmark with ground truth labels. The batch size is equal to 64.

Method	LR value	CIFAR10-C (L)	ImageNet-C (L)	ImageNet-R (L)	DomainNet-126 (L)	Avg.
AdaContrast	Default	81.8	18.8	26.5	61.7	47.2
	Transfer IN-C	81.2	36.1	40.8	59.7	54.5
I-AdaContrast	Default	85.4	40.4	38.2	64.4	57.1 ^{+9.9}
	Transfer IN-C	85.4	40.4	38.2	64.4	57.1 ^{+2.6}
CoTTA	Default	56.0	52.8	50.5	45.6	51.2
	Transfer IN-C	11.2	52.8	50.5	45.6	40.0
I-CoTTA	Default	68.4	35.4	39.6	56.8	50.1 ^{-1.1}
	Transfer IN-C	52.0	35.4	39.6	56.8	46.0 ^{+6.0}
RoTTA	Default	82.3	13.2	43.4	50.3	47.3
	Transfer IN-C	73.2	30.8	41.0	55.3	50.1
I-RoTTA	Default	79.6	32.7	39.7	57.2	52.3 ^{+5.0}
	Transfer IN-C	79.3	33.3	39.9	57.2	52.4 ^{+2.3}

A.10 POTENTIAL OF ADAPTIVE β VALUE.

In Table A.5, we explore a dynamic approach to adjusting the teacher model’s momentum parameter (β). Our experiment begins with the default value of $\beta = 0.999$, allowing initial teacher model plasticity, then transitions to complete weight preservation of IT ($\beta = 1.0$) after one full cycle through the data. This hybrid approach outperforms our IT technique in several cases, demonstrating the potential of adaptive momentum strategies.

However, the results are not uniformly positive with our standard IT outperforming the hybrid method in some cases (AdaContrast on ImageNet-C (L) and CoTTA on DomainNet-126 (L)). This suggests that the fixed period length is not a universal value and there is a need to adjust it correctly.

Table A.5: Classification accuracy [%] for long scenarios with the weights of the teacher fixed only after the 1st loop on the test sequence. The value in superscript indicates the improvements over the IT technique’s performance. The batch size is equal to 64.

Method	CIFAR10-C (L)	ImageNet-C (L)	ImageNet-R (L)	DomainNet-126 (L)	Avg.
AdaContrast	85.2 ^{-0.1}	38.4 ^{-2.0}	38.2 ^{+0.1}	65.3 ^{+0.9}	56.8 ^{-0.3}
CoTTA	72.0 ^{+3.7}	45.0 ^{+9.6}	42.8 ^{+3.3}	49.1 ^{-6.9}	52.2 ^{+2.4}
RoTTA	80.4 ^{+0.7}	36.1 ^{+3.2}	41.0 ^{+1.3}	57.9 ^{+1.3}	53.9 ^{+1.7}

A.11 DISCUSSION ON MODEL RESET MECHANISM.

CoTTA’s proposed resetting mechanism aims to preserve source knowledge by stochastically restoring portions of the student model’s weights to their original source state during each update iteration. In principle, an effective source knowledge preservation technique should eliminate the need for our IT technique.

However, CoTTA’s reset mechanism introduces a restoration probability parameter. To ensure our findings were not biased by suboptimal parameter selection, we conducted parameter tuning experiments, documented in Table A.6. These results reveal that the optimal restoration probability varies across datasets, with model performance dependent on this parameter. When following a realistic scenario of tuning on a single dataset, the performance improvements were marginal (Avg. Transfer IN-C). Only by using an Oracle approach on all benchmarks, we observe performance gains, highlighting the practical limitations of this approach.

Table A.6: Classification accuracy [%] for long scenarios with restoration probability parameter p of CoTTA method tuned. The batch size is equal to 64. **Avg. Def.** is the average accuracy with default p value. **Avg. Transfer IN-C** is the average accuracy with a single p value chosen on the ImageNet-C dataset using the Oracle method. Average accuracy when the p value is chosen separately for each of the datasets with Oracle is presented in **Avg. Oracle** column.

p value	CIFAR10-C (L)	ImageNet-C (L)	ImageNet-R (L)	DomainNet-126 (L)	Avg.		
					Def.	Transfer IN-C	Oracle
0.1	73.1	29.0	41.8	26.9	51.2	51.6	58.1
0.01	53.7	24.8	35.6	13.7			
0.001 (Def.)	56.0	52.8	50.5	45.6			
0.0001	54.7	53.7	45.0	52.9			
0.00001	54.3	53.6	49.0	55.0			
0.0	52.7	53.5	48.9	54.5			

A.12 DISCUSSION ON RDUMB.

RDumb has already been established as a state-of-the-art baseline method for extended adaptation scenarios, demonstrating great performance in both prior work (Press et al., 2023) and our current experiments. Despite its effectiveness, limitations should be considered.

The method’s mechanism of periodically resetting the model to its initial state leads to significant accuracy drops immediately following each reset, as illustrated in Figure A.2. Such instability is particularly concerning since reliable test-time adaptation should maintain consistent performance

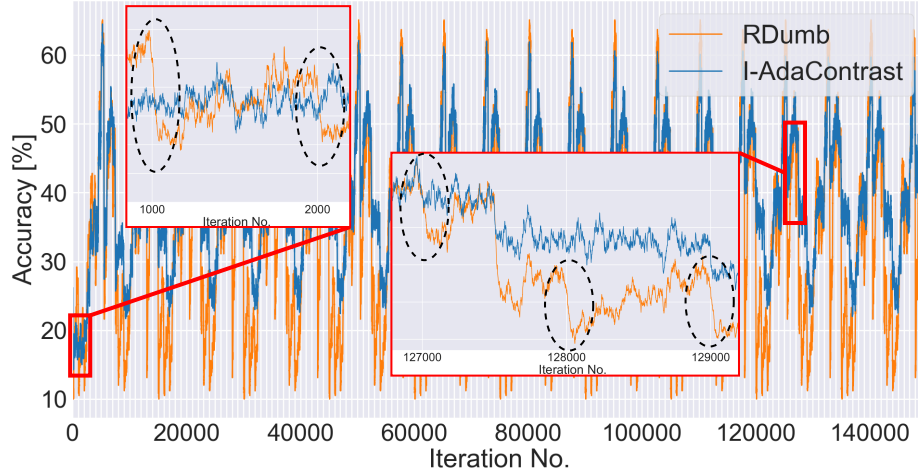


Figure A.2: Batchwise accuracy plots of RDumb and I-AdaContrast methods on ImageNet-C (L) benchmark. The accuracy values were smoothed to make the plot clearer. RDumb resets the model every 1000 iterations, which causes significant drops in accuracy after the reset.

throughout the adaptation process. Furthermore, the same constant reset interval is likely not optimal for every case, which adds a hyperparameter to select. In contrast, our IT approach achieves comparable performance without requiring parameter tuning.

A.13 ADAPTATION TO REPEATED SOURCE DOMAIN DATA.

We investigated whether the observed accuracy degradation during adaptation stems solely from distribution shift by conducting experiments on the source domain’s validation splits. We evaluated performance under two conditions: a single pass through the data (1x) and 20 repeated passes (20x), with results shown in Table A.7. Our findings reveal that accuracy degradation occurs even on source domain data, with dataset-specific variations. This phenomenon is visible on all tested datasets except CIFAR10-C. We attribute this exception to CIFAR10-C’s lower complexity, particularly its smaller number of classes compared to other datasets in our study.

The IT in most cases improves the performance on repeated streams (20x), however, the increased stability negatively impacts the accuracy on the 1x streams (especially with CoTTA and RoTTA).

Table A.7: Classification accuracy [%] for the adaptation on the source domain’s validation splits. 1x indicates the performance on a single pass through the data, while 20x means the accuracy on the 20 repeated passes. The batch size is equal to 64. The degradation of performance also occurs when adapting to the source domain, however, this effect depends on the dataset and the method used.

Method	CIFAR10-C		ImageNet-C		ImageNet-R		DomainNet-126		Avg.	
	1x	20x	1x	20x	1x	20x	1x	20x	1x	20x
AdaContrast	93.6	93.7	72.3	38.4	91.4	87.1	93.2	85.7	87.6	76.2
I-AdaContrast	93.6	93.7	72.8	66.5	91.4	88.8	94.1	92.8	88.0	85.5
CoTTA	93.5	92.9	74.2	63.2	91.7	90.2	86.1	61.2	86.4	76.9
I-CoTTA	77.4	81.6	51.0	60.5	77.5	88.1	74.1	84.7	70.0	78.7
RoTTA	94.2	94.4	75.7	63.2	91.9	81.5	89.1	58.8	87.7	74.5
I-RoTTA	94.1	93.5	73.1	72.9	90.7	85.4	68.8	88.0	81.7	85.0

A.14 ADDITIONAL RESULTS

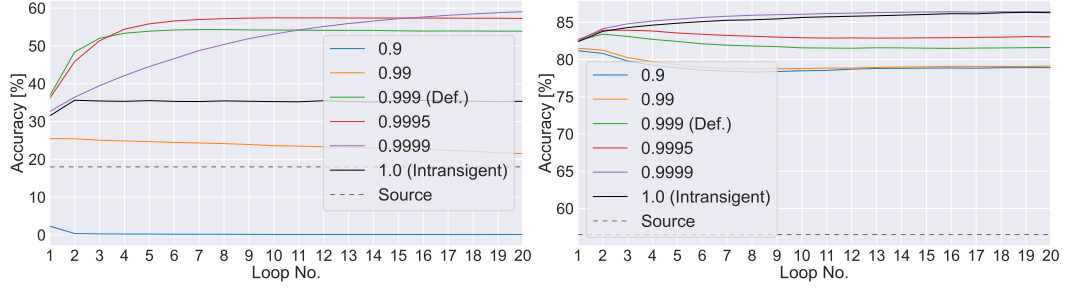


Figure A.3: Mean accuracy [%] for each loop of common testing sequence on ImageNet-C (L) using CoTTA (**left**) and on CIFAR10-C (L) using AdaContrast (**right**). The Brown dashed line indicates the Source model accuracy.

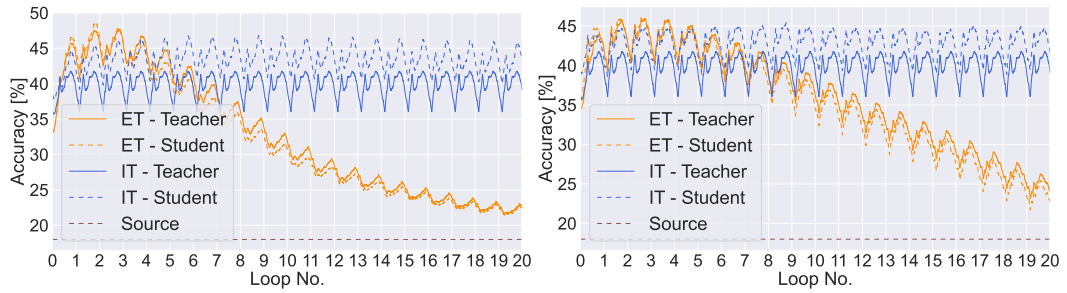


Figure A.4: Per batch accuracy [%] on ImageNet-C (L) comparing AdaContrast (**left**) and RoTTA (**right**) using ViT-B16 network with EMA teacher (ET, orange) and intransigent teacher (IT, blue), both for teacher (solid) and student (dashed).

Table A.8: Classification accuracy [%] for common length sequences.

Method	CIFAR10-C	ImageNet-C	ImageNet-R	DomainNet-126
Source	56.5	18.0	36.2	54.7
MEMO (Zhang et al., 2022)	65.6	25.0	40.9	53.2
BS = 10				
TestBN	75.0	27.0	36.6	46.5
TENT (Wang et al., 2021)	75.7	31.2	38.9	52.4
EATA (Niu et al., 2022)	77.4	36.0	43.1	54.4
SAR (Niu et al., 2023)	75.8	31.3	41.9	52.8
RDUMB (Press et al., 2023)	77.2	34.8	41.3	52.0
AdaContrast (Chen et al., 2022)	81.3	33.3	39.5	56.5
I-AdaContrast	82.0	33.8	39.8	59.6
CoTTA (Wang et al., 2022)	75.1	26.4	41.1	52.0
I-CoTTA	69.8	28.3	35.6	49.5
RoTTA (Yuan et al., 2023a)	79.0	29.2	38.6	55.9
I-RoTTA	73.2	29.4	39.3	56.6
PETAL (Brahma & Rai, 2023)	68.3	23.2	36.6	49.5
I-PETAL	74.2	27.3	36.6	49.5
BS = 64				
TestBN	79.2	31.4	39.7	54.5
TENT (Wang et al., 2021)	77.8	37.3	42.6	58.0
EATA (Niu et al., 2022)	79.8	42.0	45.8	59.7
SAR (Niu et al., 2023)	79.3	37.8	42.8	57.2
RDUMB (Press et al., 2023)	81.4	40.0	46.2	58.9
AdaContrast (Chen et al., 2022)	82.6	34.8	40.9	62.0
I-AdaContrast	82.4	35.1	41.0	61.7
CoTTA (Wang et al., 2022)	82.2	36.8	42.8	58.9
I-CoTTA	68.6	31.7	35.9	54.4
RoTTA (Yuan et al., 2023a)	80.9	32.4	39.2	56.8
I-RoTTA	76.7	30.6	39.3	56.3
PETAL (Brahma & Rai, 2023)	76.6	31.5	39.7	54.5
I-PETAL	78.4	31.4	39.7	54.5

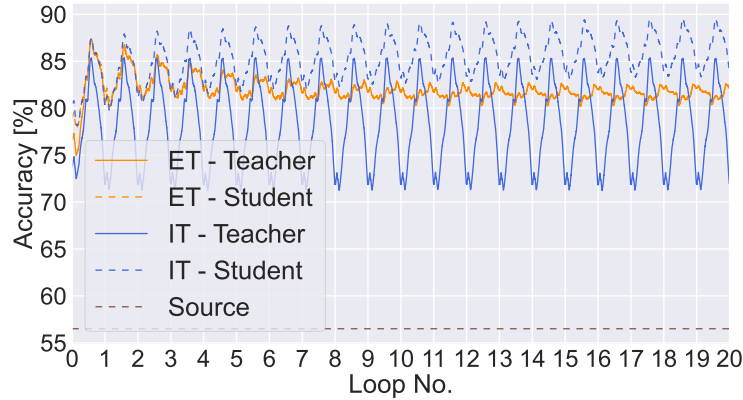


Figure A.5: Per batch accuracy [%] on CIFAR10-C (L) using AdaContrast and WideResNet-28 network with EMA teacher (ET, orange) and intransigent teacher (IT, blue), both for teacher (solid) and student (dashed).

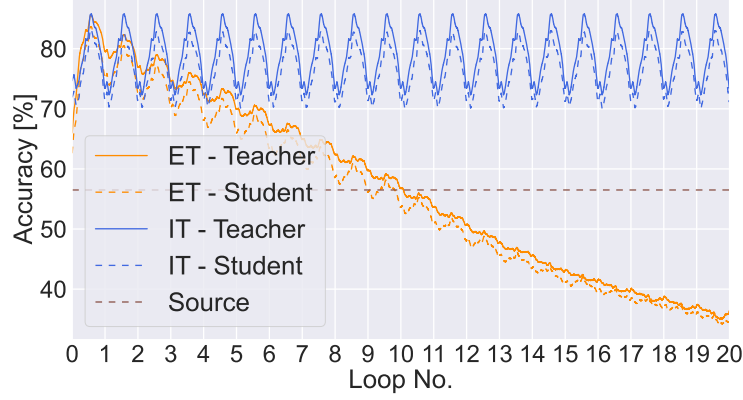


Figure A.6: Per batch accuracy [%] on CIFAR10-C (L) using CoTTA and WideResNet-28 network with EMA teacher (ET, orange) and intransigent teacher (IT, blue), both for teacher (solid) and student (dashed).

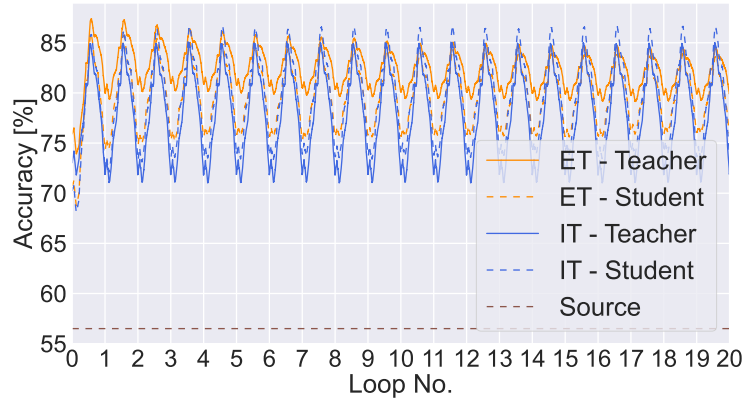


Figure A.7: Per batch accuracy [%] on CIFAR10-C (L) using RoTTA and WideResNet-28 network with EMA teacher (ET, orange) and intransigent teacher (IT, blue), both for teacher (solid) and student (dashed).