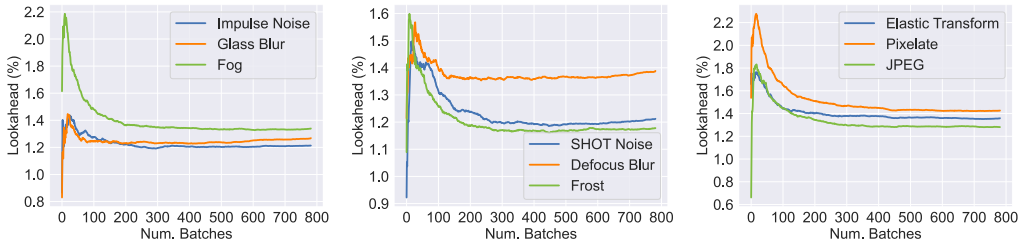
A TEST TIME ADAPTATION WITH AUXILIARY TASKS

A.1 DISTA: DISTILLATION BASED TEST TIME ADAPTATION

In Section 2.2, we showed how our proposed auxiliary task in DISTA had a positive lookahead for three corruptions from the ImageNet-C benchmark. Here, for the sake of completeness, we provide the lookahead plots for the remaining corruptions in ImageNet-C in Figure 3. We observe, similarly to our earlier findings in Section 2.2, that our auxiliary task has a consistent positive lookahead across all corruptions. That is, our distillation loss on clean data helps to better adapt to domain shifts. Note that this is already demonstrated through our extensive experimental evaluation in Sections 4.1-4.3 where DISTA consistently outperformed previous state-of-the-art TTA methods.



(a) DISTA (equation 4). (b) DISTA (equation 4). (c) DISTA (equation 4).

Figure 3: **Lookahead Analysis.** We plot the lookahead of DISTA for the 12 different corruptions from the ImageNet-C benchmark. We find that our proposed auxiliary task always yields a positive lookahead across all considered corruptions. These results corroborate our hypothesis that optimizing our distillation task on clean data helps adapting to distribution shifts.

B ADDITIONAL EXPERIMENTS

B.1 EXPERIMENTAL SETUP AND HYPER-PARAMETER CHOICES

In Section 4, we outlined our experimental setup in terms of architectures and evaluation protocols. In this section, we delve more deeply into implementation and experimental details that, due to space constraints, were not able to elaborate on in the main paper. For all baselines, we used the official code released by the authors to reproduce their results with their recommended hyperparameters. Note that all analyzed TTA methods (except SHOT) operate solely on the normalization layers of a given network. That is, θ always refers to the learnable parameters of the normalization layers (*e.g.* BatchNorm layers). Further, and following Wang et al. (2021) and Niu et al. (2022), we use an SGD optimizer with learning rate of 25×10^{-4} and momentum of 0.9. For DISTA, we follow Niu et al. (2022) in setting $\epsilon = 5 \times 10^{-2}$ in equation 4 but pick a higher value for E_0; we set $E_0 = 0.5 \log(1000)$ instead of $0.4 \log(1000)$, since we observed better lookahead with modest increases in E_0. Yet, as we as show in a later section, we still observe better results with DISTA than with EATA even when keeping $E_0 = 0.4 \log(1000)$. Regarding Aux-Tent, we set the learning rate to 5×10^{-4}. For Aux-SHOT, the learning rate is set to the default value recommended by SHOT.

Table 7: **Continual Evaluation on ImageNet-C Under Different Domain Orders.** We report the average error rate on corrupted (across all 15 corruptions) and clean domains with different random orders of domains. The first two columns are the summary of the evaluation in Section 4.2. We observe a more stable adaptation with DISTA in comparison to EATA under different domain orders where the performance gap surpasses 10%. Lower is better.

Seed	Ordered		42		4242		424242		Avg.	
	Corr.	Clean	Corr.	Clean	Corr.	Clean	Corr.	Clean	Corr.	Clean
EATA	56.5	32.7	63.6	38.5	64.7	39.4	65.8	40.4	62.7	37.8
DISTA	**50.2**	**26.3**	**52.3**	**27.6**	**52.2**	**27.7**	**52.6**	**28.2**	**51.8**	**27.4**

Table 8: **Episodic Evaluation on ImageNet-C Benchmark.** We report the results of employing parallel update (DISTA-P) compared with sequential update (DISTA) for the sake of improving efficiency. We observe that both solvers yield comparable results that are consistently better than EATA. Hence, under sifficient memory availability, one can improve latency with the parallel update.

	Noise			Blur				Weather				Digital				
	Gauss	Shot	Impul	Defoc	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contr	Elastic	Pixel	Jpeg	Avg.
EATA	64.0	62.1	62.5	66.9	66.9	52.5	47.4	48.2	54.2	40.2	32.2	54.6	42.2	39.2	44.7	51.9
DISTA	62.2	59.9	60.6	65.3	65.3	50.4	46.2	46.6	53.1	38.7	31.7	53.2	40.8	38.1	43.5	50.4
DISTA-P	62.4	60.1	61.0	65.0	65.0	50.6	46.4	46.8	53.2	39.0	31.9	53.4	41.1	38.3	43.7	50.5

B.2 Continual Evaluation

In Section 4.2, we evaluated DISTA under the continual learning setup where the stream $\mathcal{S}$ contains multiple distribution shifts presented one at a time. We followed the evaluation setup from Niu et al. (2022) regarding the order of types of domain shift in the stream $\mathcal{S}$. Here, and for completeness, we evaluate DISTA and compare it to EATA when the order of different domains is shuffled. We report the results across 3 random seeds that control the randomness of domains in $\mathcal{S}$ in Table 7.

We observe that while randomly shuffling the domains of ImageNet-C in the stream $\mathcal{S}$ has a large impact on the performance of EATA, DISTA is much more robust against such variation. That is, we report a performance drop of 7-9% for EATA when the corruptions are randomly ordered, and thus more severe shifts between presented domains are expected compared to a nicely ordered sequence. However, the same effect is virtually absent when using DISTA, for which the added randomness in domain order had little effect on the performance either on corrupted or clean domains. This brings another demonstration of the stability of DISTA under different evaluation schemes.

B.3 Analysis

B.3.1 Computational Burden

In Section 4.4.1, we discussed an alternative approach of solving the DISTA optimization problem for the sake of improving efficiency. In particular, we considered a parallel update in equation 3. We compare the performance of the alternating solver (DISTA) and parallel solver (DISTA-P) against EATA in Table 8 (Episodic evaluation on ImageNet-C). We observe the performance of DISTA-P is on par with that of DISTA, with both variants outperforming EATA by a significant margin. That is, our proposed auxiliary task is boosting the performance irrespective of the deployed solver. Hence, one can improve the efficiency (latency) by employing the parallel solver for our proposed objective in equation 4 when sufficient memory is available.

B.3.2 Ablation Studies

In Section 4.4.2, we analyzed the sensitivity of DISTA under different batch sizes when compared against Tent and EATA. We showed how DISTA is much more stable than both approaches when tested with very small batch sizes. Here, we step up the game and analyze DISTA under the smallest batch size of 1 where most TTA methods fail.

SAR (Niu et al., 2023) provided state-of-the-art results under this realistic evaluation (batch size of 1) by employing a stable update and leveraging a ViT architecture, where Layer Normalization layers are independent of the batch size. In that regard, we fix the architecture in this section to ViT

Table 9: **Episodic Evaluation on ImageNet-C Under Batch Size of 1.** We compare DISTA and SAR under batch size of 1 when emplying the ViT architecture. We observe that DISTA significantly outperforms SAR under this setting.

	Noise			Blur				Weather				Digital				
	Gauss	Shot	Impul	Defoc	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contr	Elastic	Pixel	Jpeg	Avg.
SAR	54.2	56.4	53.4	46.4	49.2	42.5	46.9	41.3	46.7	31.1	23.8	34.3	41.8	31.1	33.7	42.19
DISTA	**47.5**	**48.7**	**46.6**	**44.8**	**44.7**	**40.2**	**42.0**	**32.9**	**34.2**	**27.4**	**22.0**	**32.4**	**35.8**	**29.7**	**32.6**	**37.43**

Table 10: **Episodic Evalutation on ImageNet-C of SHOT with different auxiliary components.** We experiment with different auxiliary components when combined with SHOT. (Aux.) represents applying SHOT on both clean and corrupted data. (Fil) combines the previous approach with filtering unreliable examples. (DIS) replaces SHOT as an auxiliary task with our distillation task.

	Noise			Blur				Weather				Digital				
	Gauss	Shot	Impul	Defoc	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contr	Elastic	Pixel	Jpeg	Avg.
SHOT	73.1	69.8	72.0	76.9	75.9	58.5	52.7	53.3	62.2	43.8	34.6	82.6	46.0	42.3	48.9	59.5
+ Aux	67.1	64.9	65.7	69.0	69.9	55.5	49.8	50.7	58.7	42.3	33.3	68.2	44.4	41.1	46.5	55.1
+ Fil.	66.2	64.1	64.3	68.5	68.7	54.9	49.0	50.0	56.7	41.7	32.7	64.2	44.0	40.6	45.9	54.1
+ DIS	**64.9**	**62.6**	**62.7**	**67.1**	**66.9**	**52.9**	**47.9**	**48.6**	**55.4**	**40.5**	**32.4**	**61.8**	**42.9**	**39.3**	**44.7**	**52.7**

where we update the learnable parameters of the normalization layers. We compare the performance of SAR and DISTA under this setting and with batch size of 1 in Table 9. We observe that DISTA significantly outperforms SAR under this setup. In particular, DISTA provides an average of $\sim$ 5% reduction on the error rate under episodic evaluation on ImageNet-C. This performance gain is consistent across all corruptions in the ImageNet-C benchmark.

B.3.3 Orthogonality of Auxiliary Tasks

In Section 4.4.3, we showed how our auxiliary task approach is orthogonal to the underlying TTA method. In particular, we showed in Table 6 how applying an auxiliary task on clean data helps with either a Tent-like or a SHOT-like approach. Here we delve more onto this orthogonality. For the sake of this study, we pick SHOT as a TTA method. We report in Table 10 the effect of different auxiliary components on the overall performance of SHOT. Note that we fix the architecture to ResNet-50 and conduct episodic evaluation on the ImageNet-C benchmark.

First, we observe that employing an auxiliary task given by the SHOT objective computed on clean data improves the results significantly ($> 4\%$). Further, we combine the aforementioned approach with the filtering approach of not updating the model on unreliable examples where we observe another performance boost of 1%. At last, we replace SHOT as an auxiliary task with our proposed distillation scheme in Section 2.2, while maintaining the SHOT objective on corrupted data. In this case, we observe another significant performance boost, corroborating the superiority of our proposed auxiliary task and the orthogonality of our components to the adaptation method.

B.3.4 Components of DISTA

At last, we ablate the effect of each component of DISTA on the performance gain. Note that DISTA is reduced to EATA if we remove the proposed auxiliary task. To that end, we report in Table 11 the error rate of EATA, and its enhanced version through our proposed auxiliary task. Fist, we analyze the effect of introducing our distillation scheme via Cross Entropy (CE) on clean data without filtering. We observe a 0.5% reduction in the average error rate, with the performance gain reaching 0.8% on the motion blur corruption. Further, we analyze combining the aforementioned approach with filtering unreliable samples (by employing $\lambda_s(x_s)$), observing another 0.4% performance boost. Finally, we include sample reweighing and increase the filtering margin E_0 to $0.5\log(1000)$ resulting in another boost in accuracy (reduction in error rate). We note that we set the best hyperparameters for EATA, as recommended by the authors, with $E_0 = 0.4\log(1000)$.

Table 11: **Ablating DISTA with Episodic Evaluation on ImageNet-C.** We report the effect of each component of DISTA where (CE) represents the distillation via Cross Entropy, (Fil) represents the filtering, and DISTA is the an improved version with better hyperparameter (setting $E_0 = 0.5\log(1000)$. Note that each proposed component provides a consistent performance boost.

	Noise			Blur				Weather				Digital				
	Gauss	Shot	Impul	Defoc	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contr	Elastic	Pixel	Jpeg	Avg.
EATA	64.0	62.1	62.5	66.9	66.9	52.5	47.4	48.2	54.2	40.2	32.2	54.6	42.2	39.2	44.7	51.9
+ CE	63.2	61.2	61.6	66.3	66.3	51.7	46.9	47.9	53.9	39.7	31.9	54.3	41.9	39.1	44.4	51.4
+ Fil.	62.9	60.7	61.4	65.8	65.9	51.2	46.5	47.6	53.7	39.3	31.7	54.3	41.6	38.5	44.1	51.0
DISTA	**62.2**	**59.9**	**60.6**	**65.3**	**65.3**	**50.4**	**46.2**	**46.6**	**53.1**	**38.7**	**31.7**	**53.2**	**40.8**	**38.1**	**43.5**	**50.4**