Supplementary Material for
" MetaMix: Towards Corruption-Robust Continual Learning with Temporally Self-Adaptive Data Transformation "

A. Appendix Organization

The appendix is organized as following: We first describe the baseline details. we then describe the 15 common corruption operations applied during testing. We then provide additional implementation details. We next provide additional experimental results on CIFAR10, standard deviation, robustness accuracy of class-CL and effect of memory size. We then provide additional ablation study.

B. Notation Table

We provide notation table in Table 4:

C. Baseline Description

C.1. CL backbone baselines

The backbone CL baselines are described as follows:

- **DER ++** [5] is a memory-based approach and is one of SOTA CL baselines. Our method is orthogonal to memory-based CL methods and can be seamlessly and straightforwardly integrated with them.
- **CLS-ER** [2] is an memory-replay based SOTA CL method which maintains short-term and long-term semantic memories that interact with the episodic memory to mitigate forgetting.

C.2. Data Augmentation Baselines

The data augmentation baselines are described as follows:

- **Adversarial Training** (AT) [33], which trains the model by optimizing the model performance on the worst-case perturbed input.
- **RandAugment** (RA) [11] significantly reduces data augmentation search space and can be directly trained on the target task without resorting to a separate proxy task.
- **Maxup** [16] performs data augmentation by optimizing the mixing weights of Mixup [57] in the worst-case.
- **DeepAugment** [20] which augment new images by perturbing the representations/features of deep networks.
- **Augmix** [22] composes and combines different augmentation operations with different depths and widths to generate complex corruptions and has demonstrated the effectiveness for achieving robustness against various corruptions during testing with state-of-art performance.

D. Common Corruption Description

We provide detailed descriptions of common corruptions in Table 5.

E. Additional Implementation Details

Computing Resources : We use Nvidia-A6000 to do the experiment.

We set the inner-loop step J to 3, and the learning rate α to 0.05. For this set of experiments, we train on each CL task for 50 epochs. Following Augmix [22], we use 3 examples per Jensen-Shannon Divergence (1 clean image and 2 augmented images), a chain depth stochastically varying from 1 to 3, and 3 augmentation chains. We randomly select augmentation

Table 4. Notation Table

Notation	Meaning
A	Augmentation width
$\mathcal{C}$	a collection of corruption operations
c	a corruption operation $c \in \mathcal{C}$
d	augmentation depth
$\mathcal{D}_i^{te}$	the testing data of i^{th} task
$\boldsymbol{e}_r$ and $\boldsymbol{e}_t$	the last layer features outputted by ResNet18 for memory data and current received data
$f_{\boldsymbol{\theta}}$	the CL model with parameters $\boldsymbol{\theta}$
$\boldsymbol{g}_t$	is the cell state for each datapoint in the batch of LSTM at time t
$\boldsymbol{h}_t$	is the hidden state for each datapoint in the batch of LSTM at time t
$\boldsymbol{I}_t$	is the context information encoding as input to LSTM at time t;
J	Meta Mixer update steps
$\mathcal{L}_{\boldsymbol{\theta}_t}(\boldsymbol{x}, y)$	the loss function for labeled data $(\boldsymbol{x}, y)$
$p_{\boldsymbol{x}_b} = f_{\boldsymbol{\theta}_t}(\boldsymbol{x}_b)$	is the network output probabilities of each class for original raw data $\boldsymbol{x}_b$
m_t	mixing weight at time t for mixing original data and augmented data
N	the number of CL tasks
$\mathcal{O}$	corruption operation during training
$\boldsymbol{o}_t$	is the output of LSTM at time t for the mixing parameters
$\mathcal{S}_t$	pseudo-seen augmentation operations at time t
$\mathcal{U}_t$	pseudo-unseen augmentation operations at time t
$\mathcal{T}_k$	task k
$\boldsymbol{w} = (w_1, w_2, ..., w_A)$	the mixing weight for the chains, w_k is the mixing weight for the k^{th} chain.
$\widehat{\boldsymbol{x}}_{b1}$ and $\widehat{\boldsymbol{x}}_{b2}$	are the two mini-batch memory data augmented by applying the pseudo-unseen augmentation operations $\mathcal{U}_t$ on $(\boldsymbol{x}_b, y_b)$;
$\boldsymbol{x}'_{b1}$ and $\boldsymbol{x}'_{b2}$	are the two mini-batch memory data augmented by applying the pseudo-seen augmentation operations $\mathcal{S}_t$ on $(\boldsymbol{x}_b, y_b)$
y	data label
JS	Jensen-Shannon divergence
$\boldsymbol{\phi}$	LSTM MetaMix with parameters
α	Beta and Dirichlet distribution parameter
β	Metamixer learning rate
γ	CL model learning rate
λ	regularization weight

operations from the pseudo-seen and pseudo-unseen operations split with specific severity level and chain length at each training step.

Corruption operations splitting strategy. The operations list is the augmentation operations performed during training denoted as $\mathcal{O}$ = [autocontrast, equalize, posterize, rotate, solarize, shear-x, shear-y, translate-x, translate-y]. At each training step, we first randomly shuffle the operation list and split the operations into seen and unseen operations based on a uniform random number q between [5, 7]. This is to ensure that every operation subsets are non-empty. The operations $\mathcal{O}_{1\cdots q}$ will

Table 5. Common corruption summarization

Corruption Type	**Description**
Gaussian noise	This corruption can appear in low-lighting conditions
Shot noise	is electronic noise caused by the discrete nature of light itself
Impulse noise	is a color analogue of salt-and-pepper noise and can be caused by bit errors
Defocus blur	occurs when an image is out of focus.
Frosted Glass Blur	appears with "frosted glass" windows or panels.
Motion blur	appears when a camera is moving quickly.
Zoom blur	occurs when a camera moves toward an object rapidly.
Snow	is a visually obstructive form of precipitation.
Frost	forms when lenses or windows are coated with ice crystals.
Fog	shrouds objects and is rendered with the diamond-square algorithm.
Brightness	varies with daylight intensity.
Contrast	can be high or low depending on lighting conditions and the photographed object's color.
Elastic	transformations stretch or contract small image regions.
Pixelation	occurs when upsampling a lowresolution image.
JPEG	is a lossy image compression format which introduces compression artifacts.

serve as the pseudo-seen operations, and $\mathcal{O}_{q+1\cdots 9}$ will serve as the pseudo-unseen operations.

F. Additional Experimental Results

F.1. Results on CIFAR10

Table 6 and 7 show the additional results on CIFAR10.

Table 6. Robust accuracy of **Task-CL** on **Split-CIFAR10-C**

Method	Noise			Blur				Weather				Digital				
	Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG	**Avg**
DER	50.97	50.76	51.08	50.47	50.12	50.57	50.53	50.25	50.18	50.12	50.29	50.16	50.35	50.42	50.45	50.45
AT	51.12	51.27	50.97	51.65	51.45	51.63	51.51	51.33	51.68	51.01	51.4	50.82	51.58	51.51	51.49	51.36
RA	54.45	54.36	54.32	54.48	56.17	54.06	53.84	52.97	53.03	54.85	54.52	53.89	54.64	54.76	55.5	54.39
DeepAugment	49.47	49.47	49.58	49.32	49.49	49.46	49.27	49.50	49.66	49.45	49.68	49.81	49.40	49.40	49.35	49.49
Maxup	47.56	47.49	47.88	47.59	47.48	47.58	47.68	47.78	47.99	48.93	47.81	48.05	47.52	47.45	47.41	47.75
Augmix	87.96	89.52	**87.92**	92.3	**82.17**	90.95	91.43	89.74	89.75	90.69	92.51	89.61	90.38	91.54	90.97	89.83
Ours	**88.47**	**90.22**	87.73	**94.12**	81.75	**92.36**	**93.56**	**91.04**	**90.64**	**92.14**	**94.40**	**90.78**	**91.92**	**93.26**	**92.82**	**91.01**

Table 7. Robust accuracy of **Class-CL** on **Split-CIFAR10-C**

Method	Noise			Blur				Weather				Digital				
	Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG	**Avg**
DER	15.03	14.59	15.09	11.86	14.18	12.0	11.36	13.01	11.22	10.81	13.66	11.36	12.14	13.6	13.48	12.89
AT	10.38	10.26	10.75	10.17	10.21	10.12	10.25	10.0	9.96	9.98	9.99	9.96	10.14	10.18	10.19	10.17
RA	12.59	12.55	12.02	10.65	12.27	10.8	10.51	10.34	10.41	11.17	10.42	11.48	10.66	10.79	11.02	11.18
DeepAugment	10.08	10.10	10.06	10.10	10.12	10.08	10.16	9.98	9.98	10.02	10.02	9.99	10.09	10.08	10.11	10.07
Maxup	10.04	10.04	10.04	10.04	10.04	10.04	10.03	10.04	10.04	10.04	10.04	10.04	10.04	10.04	10.04	10.04
Augmix	58.01	61.64	58.33	69.17	48.06	66.28	67.69	62.31	**63.85**	**66.07**	**70.83**	**64.54**	63.96	66.02	64.61	63.42
Ours	**59.60**	**63.78**	**58.76**	69.14	**51.31**	**66.85**	**67.78**	**62.85**	63.63	65.77	70.53	61.89	**65.63**	**66.86**	**65.12**	**63.97**

F.2. Standard Deviation of Robustness Accuracy

The standard deviation on CIFAR100 with task- and class-CL are shown in Table 8 and 9.

Table 8. Standard Deviation of Robust accuracy of **Task-CL** on **Split-CIFAR100-C** with **DER++**

Method	Noise			Blur				Weather				Digital				
	Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG	**Avg**
DER++	0.27	0.31	0.33	0.26	0.20	0.38	0.20	0.22	0.22	0.29	0.28	0.31	0.39	0.34	0.24	0.33
AT	0.42	0.30	0.32	0.27	0.27	0.38	0.37	0.42	0.32	0.31	0.37	0.49	0.37	0.38	0.29	0.42
RA	0.97	0.80	0.67	0.87	0.94	0.54	0.52	0.92	0.68	0.81	0.87	0.84	0.81	0.98	0.66	0.54
DeepAugment	0.20	0.36	0.33	0.17	0.19	0.39	0.39	0.40	0.14	0.22	0.19	0.14	0.35	0.18	0.18	0.17
Maxup	0.15	0.20	0.12	0.26	0.19	0.26	0.38	0.16	0.25	0.10	0.16	0.25	0.23	0.27	0.24	0.35
Augmix	0.77	0.40	0.78	0.74	0.71	0.54	0.73	0.59	0.48	0.70	0.40	0.46	0.48	0.51	0.64	0.67
Ours	0.79	0.82	0.92	0.42	1.01	0.38	0.22	0.51	0.12	0.58	0.40	0.35	0.08	0.49	0.21	0.53

Table 9. Standard Deviation of Robust accuracy of **Class-CL** on **Split-CIFAR100-C** with **DER++**

Method	Noise			Blur				Weather				Digital				
	Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG	**Avg**
DER++	0.08	0.19	0.20	0.20	0.08	0.14	0.10	0.20	0.18	0.07	0.08	0.15	0.18	0.19	0.09	0.07
AT	0.12	0.15	0.07	0.10	0.12	0.11	0.19	0.11	0.06	0.18	0.16	0.12	0.18	0.17	0.16	0.14
RA	0.57	0.58	0.44	0.58	0.59	0.31	0.39	0.66	0.56	0.53	0.50	0.50	0.67	0.48	0.35	0.67
DeepAugment	0.23	0.07	0.17	0.18	0.24	0.12	0.13	0.10	0.24	0.19	0.06	0.19	0.05	0.11	0.19	0.09
Maxup	0.22	0.21	0.17	0.07	0.07	0.11	0.20	0.12	0.07	0.10	0.18	0.09	0.21	0.14	0.07	0.20
Augmix	1.19	1.11	0.81	0.73	1.12	0.85	1.09	1.15	0.71	0.73	0.89	1.16	0.83	1.03	0.76	0.89
Ours	1.16	1.27	0.78	0.83	0.92	0.95	0.79	0.94	1.12	1.11	0.93	1.17	0.81	1.06	0.88	0.96

F.3. Robust accuracy of Class-CL

Table 3 (main text) and 10 show the results of class-CL on Split-CIFAR100-C and Split-MiniImageNet-C.

Table 10. Robust accuracy of **Class-CL** on **Split-MiniImageNet-C**

Method	Noise			Blur				Weather				Digital				
	Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG	**Avg**
DER++	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01
AT	1.01	1.11	0.97	1.21	1.19	1.2	1.18	1.32	1.16	1.03	1.15	0.77	1.2	1.01	1.19	1.11
RA	2.25	2.63	2.13	3.34	2.32	2.96	3.24	3.21	2.07	2.68	2.78	2.34	3.71	3.52	4.41	2.91
DeepAugment	0.86	0.81	0.93	0.94	0.94	0.92	0.95	1.00	1.02	1.01	1.02	1.04	0.87	0.75	0.80	0.92
Maxup	1.84	1.14	1.91	1.32	1.96	1.26	1.38	1.9	1.25	1.5	1.2	0.78	1.34	1.33	1.31	1.43
Augmix	14.88	16.62	13.36	18.73	16.35	19.63	17.64	15.84	17.52	16.71	19.94	10.99	19.86	14.47	20.39	16.86
Ours	**14.91**	**17.02**	**14.15**	**21.73**	**18.69**	**23.67**	**20.74**	**16.96**	**17.85**	**19.81**	**22.50**	**12.92**	**24.35**	**15.68**	**24.40**	**19.02**

F.4. Effect of Memory Size

We evaluate the effect of memory size with 500 and 3000, respectively. The memory size of 500 is the default setting in the above tables. We provide experiment results on CIFAR100, Mini-ImageNet with memory size 3000 on task-CL and class-CL respectively in Table 11, 12, 13, 14.

Table 11. **Task-CL** with memory size 3000 on robust accuracy for various corruptions and compared methods on **Split-MiniImageNet-C**.

Method	Noise			Blur				Weather				Digital				
	Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG	**Avg**
DER++	10.47	10.44	10.31	10.17	10.34	10.26	10.30	10.49	10.45	10.24	10.28	10.10	10.40	10.37	10.39	10.34
AT	10.03	10.19	10.04	10.29	10.28	10.29	10.25	10.37	10.20	10.16	10.05	9.70	10.30	10.38	10.26	10.19
RA	20.61	21.39	20.88	18.59	18.61	18.76	17.98	19.82	20.53	16.92	18.93	14.02	21.88	22.31	26.47	19.85
DeepAugment	9.42	9.43	9.41	9.52	9.52	9.50	9.44	9.39	9.52	9.68	9.27	9.29	9.49	9.52	9.39	9.45
Maxup	9.63	9.76	9.58	9.89	9.91	9.78	9.92	10.08	10.09	9.80	9.79	9.53	9.87	10.18	9.70	9.83
Augmix	53.60	59.28	**48.45**	68.27	62.44	72.15	65.31	**64.10**	63.15	64.11	70.94	48.11	73.37	58.43	74.79	63.10
Ours	**53.71**	**60.42**	46.10	**71.65**	**64.56**	**74.12**	**68.40**	63.63	**65.23**	**68.10**	**72.23**	**51.12**	**75.06**	**60.44**	**76.15**	**64.73**

Table 12. **Class-CL** with memory size 3000 on robust accuracy for various corruptions and compared methods on **Split-MiniImageNet-C**.

Method	Noise			Blur				Weather				Digital				
	Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG	**Avg**
DER++	1.21	1.30	1.09	1.0	1.11	1.12	1.03	1.22	1.23	1.02	1.09	1.07	1.18	0.89	1.23	1.12
AT	1.05	1.09	1.03	1.18	1.20	1.16	1.15	1.02	1.01	1.02	1.02	1.01	1.15	1.20	1.14	1.09
RA	3.22	3.43	3.20	2.67	2.46	2.65	2.50	2.66	2.78	2.14	2.80	1.53	3.22	3.74	4.34	2.89
DeepAugment	1.08	1.09	1.07	1.15	1.17	1.16	1.13	1.06	1.04	1.05	1.04	1.04	1.11	1.15	1.12	1.10
Maxup	0.82	0.81	0.84	0.92	0.93	0.89	0.92	1.01	0.99	1.01	1.07	1.03	0.91	0.79	0.92	0.92
Augmix	**23.89**	27.86	**20.59**	32.07	28.18	35.71	29.84	27.59	27.14	27.29	34.98	18.66	37.14	28.10	38.96	29.20
Ours	23.28	**27.95**	19.90	**37.28**	**30.73**	**39.54**	**34.40**	**27.99**	**29.89**	**33.06**	**36.95**	**21.28**	**40.51**	**28.43**	**41.86**	**31.54**

Table 13. **Task-CL** with memory size 3000 on robust accuracy for various corruptions and compared methods on **Split-CIFAR100-C**.

Method	Noise			Blur				Weather				Digital				
	Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG	**Avg**
DER++	10.85	10.87	10.87	10.45	11.01	10.46	10.36	10.72	10.48	10.34	10.77	10.41	10.59	10.93	10.68	10.65
AT	11.23	11.27	10.98	11.70	11.28	11.69	11.78	10.86	10.93	11.20	11.52	11.79	11.59	11.50	11.38	11.38
RA	24.52	23.54	24.2	22.46	21.47	22.08	21.32	22.09	24.22	19.82	22.06	16.16	25.46	25.56	28.59	22.90
DeepAugment	12.00	12.08	11.81	12.08	11.92	12.14	12.03	11.72	11.45	11.51	11.73	10.73	11.96	12.06	12.10	11.82
Maxup	10.98	10.99	10.75	11.21	11.19	11.25	11.11	11.09	11.17	11.04	10.90	10.78	11.24	11.17	11.15	11.07
Augmix	67.34	71.50	66.41	79.13	**68.70**	77.62	77.99	74.88	73.63	74.31	79.39	73.21	76.25	77.09	74.27	74.11
Ours	**68.71**	**72.67**	**67.75**	**80.90**	68.42	**79.06**	**79.83**	**75.32**	**75.42**	**75.94**	**81.67**	**76.12**	**78.00**	**78.22**	**75.28**	**75.56**

Table 14. **Class-CL** with memory size 3000 on robust accuracy for various corruptions and compared methods on **Split-CIFAR100-C**.

Method	Noise			Blur				Weather				Digital				
	Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG	**Avg**
DER++	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
AT	1.20	1.18	1.16	1.10	1.19	1.09	1.07	1.05	1.02	1.03	1.10	1.04	1.12	1.14	1.09	1.11
RA	5.14	5.65	4.08	4.52	3.22	4.6	3.74	3.23	4.52	3.56	4.65	2.9	4.68	4.78	5.11	4.29
DeepAugment	1.42	1.44	1.43	1.46	1.49	1.46	1.45	1.32	1.22	1.20	1.26	1.07	1.44	1.46	1.47	1.37
Maxup	0.97	1.03	1.02	1.04	1.10	1.00	1.21	0.89	0.88	0.89	0.87	0.88	1.05	1.01	1.01	0.99
Augmix	27.84	31.32	27.35	39.37	28.35	37.14	38.29	34.19	33.65	35.10	39.92	33.00	35.38	38.06	33.50	34.16
Ours	**30.06**	**33.54**	**29.57**	**41.59**	**30.57**	**39.36**	**40.51**	**36.41**	**35.87**	**37.32**	**42.14**	**35.22**	**37.60**	**40.28**	**35.72**	**36.38**

F.5. BWT

In the corruption-robust scenario, backward transfer (BWT) is no longer a meaningful metric here with such extreme differences of accuracy, as the significantly lower accuracy of comparison methods results in much less space for further performance variations during backward transfer. The results are shown in Table 15.

Table 15. Various methods with **Backward Transfer (BWT)**.

Corruption	Split-CIFAR10-C		Split-CIFAR100-C		Split-miniImageNet-C	
	Task-CL	Class-CL	Task-CL	Class-CL	Task-CL	Class-CL
DER	-1.55	-46.36	-0.74	-0.69	-0.18	-1.68
AT	1.86	-28.43	-0.54	-3.83	-1.08	-4.54
RA	1.50	-44.80	2.95	-11.03	-3.52	-12.58
DeepAugment	-0.61	-25.40	-22.80	-72.79	-18.02	-59.14
Maxup	-2.84	-49.63	-0.29	0.12	-0.07	-1.12
Augmix	-5.72	-29.94	-19.70	-61.09	-16.78	-50.85
Ours	-5.03	-28.64	-19.67	-59.96	-15.93	-49.58

G. Integration with other CL methods

In this Section, we integrate our methods with another SOTA CL methods, i.e., memory-based method, CLS-ER (only applicable in Class-CL) [2] in Table 16 and 17.

G.1. CLS-ER

Table 16. Robust accuracy of **Class-CL** on **Split-CIFAR100-C** with **CLS-ER**

Method	Noise			Blur				Weather				Digital				
	Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG	**Avg**
CLS-ER	0.82	0.84	0.96	0.85	0.9	0.8	0.7	0.92	1.04	0.97	0.76	0.97	0.8	0.86	0.8	0.87
AT	1.19	1.16	1.09	1.04	1.12	0.98	1.0	0.84	1.13	1.08	0.92	1.15	1.11	1.08	1.0	1.06
RA	6.63	6.67	6.46	7.05	5.1	4.95	6.29	5.75	6.67	8.07	7.22	7.65	6.03	7.6	7.88	6.67
DeepAugment	2.66	2.44	2.54	2.55	1.81	1.98	1.81	2.04	2.17	1.89	1.74	2.4	2.24	2.64	2.17	2.20
Maxup	1.53	1.63	1.74	1.58	1.38	1.48	1.63	1.61	1.6	1.51	1.71	1.32	1.46	1.41	1.58	1.55
Augmix	19.81	21.35	20.46	26.45	19.23	25.28	25.68	22.29	22.73	23.05	26.57	22.83	23.81	24.91	23.29	23.18
Ours	22.69	24.63	23.38	29.51	22.39	28.33	28.45	25.65	25.34	26.27	29.88	26.06	27.15	28.34	26.2	26.28

Table 17. Robust accuracy of **Class-CL** on **Split-MiniImageNet-C** with **CLS-ER**

Method	Noise			Blur				Weather				Digital				
	Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG	**Avg**
CLS-ER	1.03	1.09	1.04	1.09	1.03	1.05	1.02	1.02	1.03	1.07	1.02	1.1	1.05	1.05	1.05	1.05
AT	1.07	1.23	0.99	1.28	1.3	1.22	1.21	1.34	1.32	1.14	1.17	0.81	1.32	1.09	1.31	1.19
RA	2.71	3.35	2.72	3.74	2.79	3.5	3.54	3.79	2.75	3.4	3.66	2.74	4.52	3.93	5.14	3.49
DeepAugment	2.07	1.75	1.93	1.73	1.49	1.42	1.45	2.37	1.88	2.14	2.48	2.53	2.02	1.27	1.53	1.87
Maxup	2.08	1.59	2.19	1.69	2.54	1.73	1.91	2.12	1.67	1.76	1.54	1.1	1.66	1.7	1.76	1.80
Augmix	15.14	16.77	13.66	18.98	16.5	19.78	17.79	16.12	17.72	16.86	20.01	11.06	19.95	14.65	20.61	17.04
Ours	15.1	17.23	14.46	21.95	19.01	23.8	21.12	17.18	18.12	20.1	22.84	13.24	24.57	15.97	24.63	19.29

H. Clean ACC and Unknown vs. Known Corruptions

The clean ACC on CIFAR10 is shown in the following table. All the baseline methods has some trade-off the clean accuracy since the data augmentation changes the memory data distribution. But our method is substantially more robust to data corruptions than baselines.

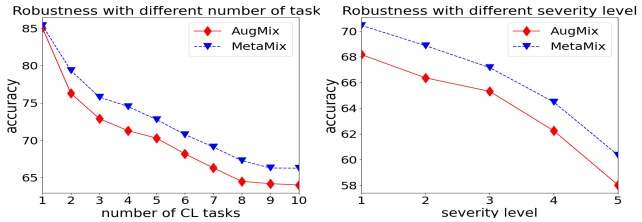
DER++	AT	RA	DeepAugment	Maxup	AugMix	MetaMix
93.56	86.73	90.83	83.28	91.07	89.39	89.51

Performance gap between known and unknown corruption. The results of known vs. unknown corruptions on CIFAR10 are shown in the following table. The gap in class-CL is larger than task-CL

	task-CL	class-CL
known corruptions	93.49	69.78
unknown corruptions	91.01	63.97

I. Ablation study

Robustness Varies Along the CL Process (time) and Across Different Severity Levels. Figure 3 in Appendix shows the robustness varies along the CL process and across different severity levels. We observe that our method still outperforms AugMix with different number of training tasks during CL. Also, the results show that the higher the corruption severity, the lower the robustness accuracy. This aligns with our intuition that with more corruptions, it will be more difficult for the CL learner to make correct predictions.



(a) Robustness varies along the CL process

(b) Robustnesss varies across different severity levels

Figure 3. (a) The robustness varies with different number of tasks, the robustness accuracy is evaluated at the end of training each task. (b) Robustness varies with different severity levels.

In this section, we perform various ablation studies to show the sensitivity of hyperparameters and the effectiveness of each proposed component.

Effect of meta-learning objectives. In the main text, we formulate the problem as a bi-level optimization in Eq. (6). Here, we explore another alternative objective with simultaneous optimization on all the training augmentation operations without splitting the operation set $\mathcal{O}$ into seen and unseen ones without the meta-learning objective. The optimization goal is shown in Eq. (9). The result is shown in Table 18. We can observe that our meta-learning-based method can improve over this joint training baseline by more than 1.5% in most cases for both task-CL and class-CL on both Split-CIFAR100C and mini-ImageNetC, showing the effectiveness of the meta-learning objective for optimizing the pseudo unseen augmentation operation performance.

$$\boldsymbol{\theta}_*, \boldsymbol{\phi}_* = \arg\min_{\boldsymbol{\phi},\boldsymbol{\theta}} \mathbb{E}_{op\in\mathcal{O}} \mathcal{L}(\boldsymbol{x}_b, y_b, \boldsymbol{\theta}, \boldsymbol{\phi}) + \lambda JS(\boldsymbol{x}_b, \widetilde{\boldsymbol{x}}_{b1}, \widetilde{\boldsymbol{x}}_{b2}), \tag{9}$$

where $\widetilde{\boldsymbol{x}}_{b1}$ and $\widetilde{\boldsymbol{x}}_{b2}$ are the augmented images by using the operations in $\mathcal{O}$.

Table 18. Ablation study of Task-CL and Class-CL with/without meta-learning.

Task-CL	Method	CIFAR100-C	Mini-ImageNet
	(Joint training)	65.49	53.57
	Ours(meta-learning)	**66.43**	**55.32**
Class-CL	Method	CIFAR100-C	Mini-ImageNet
	Ours(Joint training)	24.41	17.53
	Ours(meta-learning)	**26.05**	**19.02**

The effect of LSTM vs MLP. To evaluate the advantage that LSTM can capture the information of previous tasks with additional hidden state, we compare it to MLP (multi-layer perceptron). We present the results in Table 19. We can observe that the performance improved slightly with LSTM as MetaMixer.

Table 19. Ablation study of Task-CL and Class-CL with LSTM vs MLP respectively.

	MetaMixer Architecture	CIFAR100-C	Mini-ImageNet
Task-CL	MLP	66.27	55.09
	LSTM	**66.43**	**55.32**
	MetaMixer Architecture	CIFAR100-C	Mini-ImageNet
Class-CL	MLP	25.83	18.79
	LSTM	**26.05**	**19.02**

Sensitivity analysis of λ. To evaluate the effectiveness of regularization strengths λ in Eq. (6). Table 20 shows the sensitivity analysis of different λ values.

Table 20. Sensitivity of λ

λ	0.0	0.5	1.0	2.0
task-CL-CIFAR10	50.45	90.05	91.01	90.32

Effect of MetaMixer adaptation steps J. To evaluate the effectiveness of different number of adaptation steps J for solving the lower-level optimization in Eq. (6). Table 21 shows the sensitivity analysis of different adaptation steps J. We can observe that the performance improves slightly with more inner-loop adaptation steps. This shows that the inner-loop optimization is more accurate with more adaptation steps, thus providing a more informative signal for outer-loop augmentation strategies optimization. For efficiency, we choose $J = 3$.

Table 21. Sensitivity analysis of adaptation steps J of Task-CL on **Split-CIFAR100-C.**

$J = 1$	65.03
$J = 3$	66.43
$J = 5$	66.49

Effect of MetaMixer learning rate β. To evaluate the effectiveness of MetaMixer learning rate β for solving the lower-level optimization in Eq. (6). Table 22 shows the sensitivity analysis of different MetaMixer learning rate β.

Table 22. Sensitivity analysis of MetaMixer learning rate β of Task-CL on **Split-CIFAR100-C.**

$\beta = 0.01$	65.87
$\beta = 0.05$	66.43
$\beta = 0.2$	66.32

Effect of combining current mini-batch data received at time t **of the current task with the mini-batch data sampled from memory buffer**. In the main text, we denote the mini-batch data $(\boldsymbol{x}_r, y_r)$ randomly sampled from the memory buffer concatenated with the current received mini-batch data $(\boldsymbol{x}_t, y_t)$ together as $(\boldsymbol{x}_b, y_b)$. Table 23 shows the results of using only the mini-batch data sampled from the memory buffer, i.e., without using the mini-batch data received at time t. We can observe that with memory mini-batch data $(\boldsymbol{x}_r, y_r)$ alone, the performance of the proposed method drops to 60.6%, indicating the effectiveness of using additional current mini-batch data.

Table 23. Ablation study of Task-CL on **Split-CIFAR100-C** without using mini-batch data from current task

Ours (memory data only)	60.6
Ours (combine)	**66.43**

Effect of mixture width A. To evaluate the effectiveness of mixture width, we perform evaluations across different mixture widths A. Table 24 shows that with the increased augmentation width A, the performance increases as well. This indicates

that the augmented data becomes more complex and diverse with a wider augmentation path and is helpful for performance optimization. For computation efficiency, we use $A = 3$ for simplicity.

Table 24. Sensitivity analysis of augmentation width A of Task-CL on **Split-CIFAR100-C**

$A = 1$	64.87
$A = 3$	66.43
$A = 5$	66.51

Computation Cost. To evaluate the computation cost of Meta-Mix compared to AugMix, we set the training time of AugMix as unit 1. We compare the relative computation cost compared to AugMix in Table 25. This shows that our methods add a slightly more additional computation cost. In future work, we will improve its computation efficiency.

Table 25. Computation cost (wall clock running time for one epoch training) comparisons of Task-CL on **Split-CIFAR100-C**

Method	running time (seconds)
AugMix	91
Meta-Mix (Ours)	179