

## 2 A Reproducibility Checklist

3 To ensure that all results are easily reproducible, we use the reproducibility framework, the Machine  
4 Learning Reproducibility Checklist version 1.2, Mar.27 2019.

- 5 1. For all **models** and **algorithms** presented,
  - 6 (a) **A clear description of the mathematical setting, algorithm, and/or model.**
  - 7 (b) **An analysis of the complexity (time, space, sample size) of any algorithm.**  
8 We analyse the complexity of the proposed methods in Table 2.
  - 9 (c) **A link to a downloadable source code, with specification of all dependencies, in-**  
10 **cluding external libraries.**  
11 We provide a downloadable source code, with specification of all dependencies, includ-  
12 ing external libraries. See <https://github.com/MinghuiChen43/CIL-ReID>.
- 13 2. For any **theoretical claim**,
  - 14 (a) **A statement of the result.**  
15 We do not make theoretical claim.
  - 16 (b) **A clear explanation of any assumptions.**  
17 We do not make theoretical claim.
  - 18 (c) **A complete proof of the claim.**  
19 We do not make theoretical claim.
- 20 3. For all **figures** and **tables** that present empirical results,
  - 21 (a) **A complete description of the data collection process, including sample size.**  
22 We do not collect new dataset.
  - 23 (b) **A link to downloadable version of the dataset or simulation environment.**  
24 We do not collect new dataset and create new simulation environment.
  - 25 (c) **An explanation of any data that were excluded, description of any pre-processing**  
26 **step.**  
27 We did not exclude any data. The pre-processing steps for evaluating corruption  
28 robustness are described in Appendix.
  - 29 (d) **An explanation of how samples were allocated for training/ validation/ testing.**  
30 We provide detailed allocation of training/ validation/ testing in Appendix.
  - 31 (e) **The range of hyper-parameters considered, method to select the best hyper-**  
32 **parameter configuration, and specification of all hyper-parameters used to gen-**  
33 **erate results.**  
34 We provide detailed settings of selecting hyper-parameter in Appendix.
  - 35 (f) **The exact number of evaluation runs.**  
36 Ten evaluation runs are conducted for the experiments on Market-1501, DukeMTMC,  
37 CUHK03 (detected), SYSU-MM01 and RegDB datasets. Three evaluation runs are  
38 conducted for the experiments on MSMT17 dataset.
  - 39 (g) **A description of how experiments were run.**  
40 We provide a description of how experiments were run in Appendix.
  - 41 (h) **A clear definition of the specific measure of statistics used to report results.**  
42 We use CMC (cumulative match characteristic), mAP (mean average precision), and  
43 mINP (mean inverse negative penalty) to evaluate the performance of a ReID system.  
44 The definition of mINP is described in Sec 3.1.
  - 45 (i) **Clearly defined error bars.**
  - 46 (j) **A description of results with central tendency (e.g. mean) & variation (e.g. std-**  
47 **dev).**  
48 We report mean for all experiments. And additionally, we report standard deviation for  
49 the proposed baseline in Appendix.
  - 50 (k) **A description of the computing infrastructure used.**  
51 We present the description of the computing infrastructure used in Appendix.

Table 1: Details of the investigated methods.

Method	abbreviation	backbone	evaluated datasets	source code
Hao Luo et al. (2019)	BoT	ResNet-50	Market-1501, DukeMTMC, CUHK03, MSMT-17	<a href="https://github.com/michuanhaohao/reid-strong-baseline">https://github.com/michuanhaohao/reid-strong-baseline</a>
Mang Ye et al. (2020)	AGW	ResNet-50	Market-1501, DukeMTMC, CUHK03, MSMT-17, SYSU-MM01, RegDB	<a href="https://github.com/mangye16/ReID-Survey">https://github.com/mangye16/ReID-Survey</a>
Shuting He et al. (2021)	TransReID	ViT	Market-1501, DukeMTMC, CUHK03, MSMT-17	<a href="https://github.com/heshuting555/TransReID">https://github.com/heshuting555/TransReID</a>
Tianlong Chen et al. (2019)	ABD-Net	ResNet-50	Market-1501, DukeMTMC, MSMT17	<a href="https://github.com/VITA-Group/ABD-Net">https://github.com/VITA-Group/ABD-Net</a>
Binghui Chen et al. (2019)	MHN	ResNet-50	Market-1501, DukeMTMC	<a href="https://github.com/chenbinghui1/MHN">https://github.com/chenbinghui1/MHN</a>
Kaiyang Zhou et al. (2019)	OS-Net	OS-Net	Market-1501, DukeMTMC, CUHK03, MSMT17	<a href="https://github.com/KaiyangZhou/deep-person-reid">https://github.com/KaiyangZhou/deep-person-reid</a>
Ben Xie et al. (2020)	PLR-OS	OS-Net	Market-1501, DukeMTMC, CUHK03	<a href="https://github.com/AI-NERC-NUPT/PLR-OSNet">https://github.com/AI-NERC-NUPT/PLR-OSNet</a>
Fabian Herzog et al. (2021)	LightMBN	OS-Net	Market-1501, CUHK03	<a href="https://github.com/jixunbo/LightMBN">https://github.com/jixunbo/LightMBN</a>
Yifan Sun et al.(2017)	PCB	ResNet-50	Market-1501, DukeMTMC, CUHK03	<a href="https://github.com/syfafterzy/PCB_RPP_for_reID">https://github.com/syfafterzy/PCB_RPP_for_reID</a>
Guanshuo Wang et al.(2018)	MGN	ResNet-50	Market-1501, DukeMTMC, CUHK03	<a href="https://github.com/seathiefwang/MGN-pytorch">https://github.com/seathiefwang/MGN-pytorch</a>
Hao Luo et al. (2019)	Aligned++	ResNet-50	Market-1501, DukeMTMC, CUHK03, MSMT17	<a href="https://github.com/michuanhaohao/AlignedReID">https://github.com/michuanhaohao/AlignedReID</a>
Hyunjong Park et al. (2019)	RRID	ResNet-50	Market-1501, DukeMTMC, CUHK03	<a href="https://github.com/cvlab-yonsei/RRID">https://github.com/cvlab-yonsei/RRID</a>
Feng Zheng et al. (2018)	Pyramid	ResNet-50	Market-1501, DukeMTMC, CUHK03	<a href="https://github.com/TencentYoutuResearch/PersonReID-Pyramid">https://github.com/TencentYoutuResearch/PersonReID-Pyramid</a>
Fufu Yu et al. (2020)	Cace-Net	ResNet-50	Market-1501, DukeMTMC, MSMT17	<a href="https://github.com/TencentYoutuResearch/PersonReID-CACENET">https://github.com/TencentYoutuResearch/PersonReID-CACENET</a>
Hao Luo et al. (2019)	BoT	ResNet-50	Market-1501, DukeMTMC	<a href="https://github.com/michuanhaohao/reid-strong-baseline">https://github.com/michuanhaohao/reid-strong-baseline</a>
Dengpan Fu et al. (2020)	LUPerson	ResNet-50	Market-1501, DukeMTMC, CUHK03, MSMT17	<a href="https://github.com/DengpanFu/LUPerson">https://github.com/DengpanFu/LUPerson</a>
Zuozhuo Dai et al. (2018)	BDB	ResNet-50	Market-1501, DukeMTMC, CUHK03	<a href="https://github.com/daizuoazhuo/batch-dropblock-network">https://github.com/daizuoazhuo/batch-dropblock-network</a>
Zhedong Zheng et al. (2019)	DG-Net	ResNet-50	Market-1501, DukeMTMC, CUHK03, MSMT17	<a href="https://github.com/NVlabs/DG-Net">https://github.com/NVlabs/DG-Net</a>
Rodolfo Quispe et al. (2020)	TDB	ResNet-50	Market-1501, DukeMTMC, CUHK03	<a href="https://github.com/RQuispeC/top-dropblock">https://github.com/RQuispeC/top-dropblock</a>
Yunpeng Gong et al. (2021)	LGPR, F-LGPR	ResNet-50	Market-1501, DukeMTMC, MSMT17	<a href="https://github.com/finger-monkey/ReID_Adversarial_Defense">https://github.com/finger-monkey/ReID_Adversarial_Defense</a>

## 52 B Implementation Details

53 We conduct experiments on single-modality datasets Market-1501, CUHK03 (detected) and MSMT17,  
54 and cross-modality datasets SYSU-MM01 and RegDB. Tab. 1, we list the the methods we investigated  
55 and links of source codes we used. All the experiments are conducted on single 32G Tesla V100  
56 GPU. Detailed experimental settings are provided on [https://github.com/MinghuiChen43/](https://github.com/MinghuiChen43/CIL-ReID)  
57 CIL-ReID.

Table 2: Robustness evaluation under different corruption severity levels.

Severity	mINP	mAP	Rank-1
level-0	57.90 (0.00)	84.04 (0.00)	93.38 (0.00)
level-1	15.74 (0.44)	57.05 (0.30)	80.81 (0.40)
level-2	5.21 (0.22)	39.97 (0.40)	69.09 (0.76)
level-3	2.84 (0.17)	30.33 (0.21)	59.41 (0.77)
level-4	0.86 (0.06)	17.22 (0.33)	43.40 (0.64)
level-5	0.35 (0.05)	9.60 (0.23)	30.10 (0.71)

Table 3: Robustness evaluation under different corruption types.

type		mINP	mAP	Rank-1
Noise	Gaussian	4.34 (0.07)	31.40 (0.42)	57.65 (0.72)
	shot	4.35 (0.12)	31.15 (0.21)	57.62 (0.35)
	impulse	6.20 (0.13)	37.30 (0.36)	64.10 (0.31)
	speckle	14.30 (0.11)	48.87 (0.22)	73.64 (1.03)
Blur	defocus	15.06 (0.35)	49.57 (0.41)	73.32 (0.96)
	glass	26.02 (0.48)	60.85 (0.22)	80.56 (0.17)
	motion	19.42 (0.15)	55.87 (0.19)	77.45 (0.18)
	zoom	38.06 (0.27)	71.79 (0.01)	87.52 (0.36)
	Gaussian	15.48 (0.55)	52.20 (0.24)	75.97 (0.33)
Weather	snow	11.94 (0.29)	44.98 (0.16)	68.66 (0.58)
	frost	3.86 (0.10)	29.36 (0.23)	56.71 (0.36)
	fog	3.10 (0.05)	24.78 (0.04)	48.97 (0.67)
	brightness	23.34 (0.28)	61.18 (0.04)	82.14 (0.43)
	spatter	24.25 (0.56)	62.12 (0.51)	82.73 (0.38)
	rain	15.54 (0.15)	50.20 (0.40)	73.47 (0.49)
Digital	contrast	0.70 (0.06)	19.29 (0.06)	45.10 (0.57)
	elastic	16.30 (0.09)	47.95 (0.08)	70.70 (0.21)
	pixel	30.06 (0.03)	66.00 (0.27)	83.19 (0.13)
	JPEG compression	15.72 (0.39)	50.36 (0.24)	73.81 (0.70)
	saturate	0.48 (0.02)	18.43 (0.15)	55.58 (0.37)

## 58 C More Experimental Results

### 59 C.1 Robustness under different corruption settings

60 Tab. 2 shows the robustness under different corruption severity levels. Tab. 3 shows the robustness  
61 under different corruption types.

### 62 C.2 Detailed Results of the SOTA methods

63 In Tab. 4 and Tab. 5, we present the results of corruption robustness evaluations of the SOTA methods  
64 on datasets Market-1501, CUHK03 (detected), MSMT17, SYSU-MM01 and RegDB respectively.

## 65 D Visualization

66 In Fig. 1, we present more augmented examples under different data augmentation methods.

67 In Fig. 2, we present more comparison examples of activation maps under different data augmentation  
68 training methods.

Table 4: Corruption robustness evaluations of the SOTA methods on datasets Market-1501, CUHK03 (detected) and MSMT17.(Note that Market standards for dataset Market-1501)

Dataset	Method	Clean Eval.				Corrupted Eval.				Corrupted Query				Corrupted Gallery			
		mINP	mAP	R-1	R-5	mINP	mAP	R-1	R-5	mINP	mAP	R-1	R-5	mINP	mAP	R-1	R-5
Market	AGW	65.40	88.10	95.00	98.20	0.30	10.80	33.40	45.80	19.10	33.40	38.70	46.80	1.00	36.00	84.10	93.70
	BoT	51.00	83.90	94.30	97.80	0.10	6.60	26.20	37.90	12.90	29.10	36.70	44.70	0.30	21.70	74.30	86.30
	ABD-Net	64.72	87.94	94.98	98.37	0.26	9.81	29.65	42.49	17.81	31.10	36.02	44.15	0.60	31.32	81.38	91.89
	OS-Net	56.78	85.67	94.69	98.22	0.23	10.37	30.96	44.06	15.82	31.10	36.97	45.06	0.45	29.29	79.81	90.74
	DG-Net	61.60	86.09	94.77	97.98	0.35	9.96	31.75	44.92	16.93	29.76	35.93	43.64	1.04	34.59	83.30	93.05
	MHN	55.27	85.33	94.50	97.92	0.38	10.69	33.29	47.28	16.83	33.08	39.87	48.39	1.07	33.50	82.29	92.29
	BDB	61.78	85.47	94.63	97.80	0.32	10.95	33.79	47.84	18.93	31.58	38.42	47.48	0.70	31.82	81.40	92.19
	TransReID	69.29	88.93	95.07	98.46	1.98	27.38	53.19	68.95	34.39	52.40	58.27	68.97	4.47	52.59	88.50	96.05
	LGPR	58.71	86.09	94.51	98.34	0.24	8.26	27.72	38.56	15.24	29.52	35.82	42.65	0.69	32.51	82.93	92.45
	F-LGPR	65.48	88.22	95.37	98.31	0.23	9.08	29.35	40.86	17.97	31.21	36.16	43.47	0.78	33.98	83.76	93.00
	TDB	56.41	85.77	94.30	98.01	0.20	8.90	28.56	41.14	14.94	29.53	34.71	43.10	0.33	25.91	79.14	90.30
	LUPerson	68.71	90.32	96.32	98.81	0.29	10.37	32.22	44.53	19.68	34.16	40.09	47.20	0.79	34.35	85.68	93.63
	LightMBN	73.29	91.54	96.53	98.84	0.50	14.84	38.68	52.30	27.09	41.81	46.33	54.56	1.41	41.38	87.98	95.31
	PLR-OS	66.42	88.93	95.19	98.40	0.48	14.23	37.56	52.17	23.30	38.75	43.49	52.68	1.25	38.98	85.15	93.94
	PCB	41.97	82.19	94.15	97.74	0.41	12.72	34.93	49.88	12.66	32.35	39.72	49.30	0.85	33.03	80.89	91.55
	Pyramid	61.61	87.50	94.86	98.31	0.36	12.75	35.72	50.39	19.66	35.48	41.09	49.56	0.95	34.70	81.90	92.78
	Aligned++	47.31	79.10	91.83	96.97	0.32	10.95	31.00	45.55	14.70	29.43	35.15	44.95	0.67	30.10	75.86	89.74
	RRID	67.14	88.43	95.19	98.10	0.46	13.45	36.57	51.90	22.18	36.26	41.29	49.58	1.09	37.44	84.00	93.58
	VPM	50.09	81.43	93.79	98.19	0.31	10.15	31.17	45.23	15.48	31.75	39.23	49.11	0.71	31.06	79.42	91.15
	MGN	60.86	86.51	93.88	0.00	0.29	9.72	29.56	42.92	17.08	30.00	34.25	42.49	0.69	30.83	79.36	91.21
CUHK03	Pyramid	61.41	73.14	79.54	93.16	1.10	8.03	10.42	17.93	20.18	26.36	28.96	44.82	4.21	26.14	30.66	45.39
	MGN	51.18	62.73	69.14	88.93	0.46	4.20	5.44	10.86	10.90	14.39	15.42	28.44	1.77	15.55	18.92	32.43
	Aligned++	47.32	59.76	62.07	79.93	0.56	4.87	7.99	15.87	12.12	16.34	15.35	24.88	2.19	17.78	31.21	51.76
	RRID	55.81	67.63	74.99	91.51	1.00	7.30	9.66	17.35	18.25	23.88	26.44	42.55	3.98	24.35	28.89	44.96
	MHN	56.52	66.77	72.21	86.43	0.46	3.97	8.27	15.42	14.20	17.79	18.78	28.16	2.91	18.40	36.65	56.84
	CaceNet	65.22	75.13	77.64	90.43	2.09	10.62	17.04	28.93	25.50	31.59	30.99	43.17	6.43	30.78	51.49	71.69
	PLR-OS	62.72	74.67	78.14	91.14	0.89	6.49	10.99	20.22	19.61	25.96	25.91	37.41	5.03	26.37	45.41	65.92
MSMT17	OS-Net	4.05	40.05	71.86	82.65	0.08	7.86	28.51	40.19	1.71	18.77	35.40	45.20	0.14	16.99	59.90	74.69
	BoT	9.91	48.34	73.53	85.29	0.07	5.28	20.20	31.11	2.75	15.78	25.92	35.50	0.09	16.10	59.06	76.48
	AGW	12.38	51.84	75.21	86.30	0.08	6.53	22.77	34.08	3.82	18.42	28.06	37.33	0.15	18.08	61.45	78.43

Table 5: Corruption robustness evaluations of methods on cross-modality datasets SYSU-MM01 and RegDB.

Dataset	Method	Mode A								Mode B							
		Clean Eval.				Corrupted Eval.				Clean Eval.				Corrupted Eval.			
		mINP	mAP	R-1	R-5	mINP	mAP	R-1	R-5	mINP	mAP	R-1	R-5	mINP	mAP	R-1	R-5
SYSU-MM01	AGW	36.17	47.65	47.50	74.68	14.73	29.99	34.42	62.26	59.74	62.97	54.17	83.50	35.39	40.98	33.80	61.61
	DGTL	42.55	55.82	57.78	83.00	18.43	35.59	41.36	68.46	65.40	69.40	62.99	87.70	41.15	47.90	42.22	69.98
	HCTL	42.14	57.65	61.94	85.85	13.85	33.15	42.38	68.87	63.69	68.41	63.70	85.66	33.29	40.61	37.26	59.54
RegDB	AGW	54.10	68.82	75.78	85.24	32.88	43.09	45.44	55.26	52.40	68.15	75.29	83.74	6.00	41.37	67.54	81.23
	DGTL	61.15	75.63	85.00	91.21	37.02	48.35	53.50	62.02	55.90	73.39	82.67	89.37	6.21	46.29	75.20	87.54
	HCTL	68.15	82.46	90.63	94.66	43.77	55.83	62.09	67.78	66.50	81.05	89.17	94.27	6.80	53.79	83.63	92.82





Figure 1: Visualization of different augmented examples.

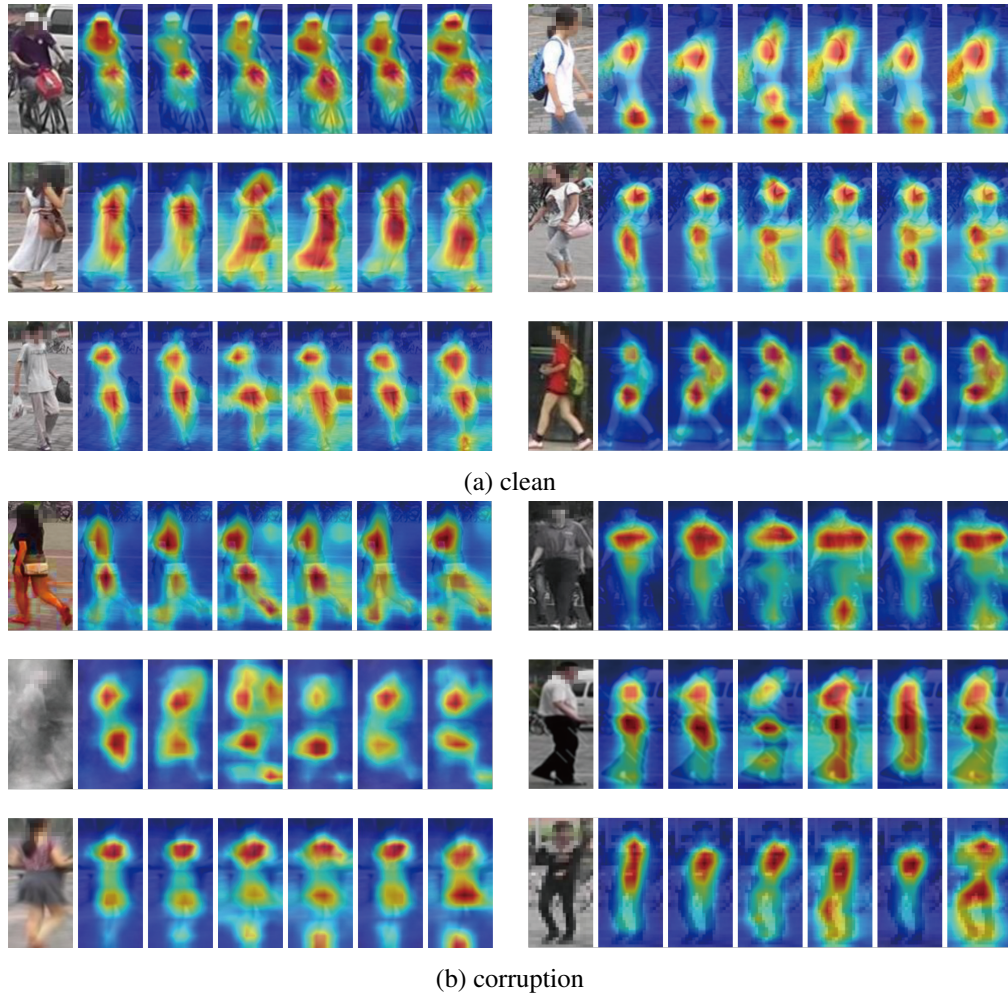


Figure 2: Visualization of activation maps. Each septet contains, from left to right, original image, activation maps from the standard model, a model trained with AugMix, random erasing, soft random erasing, random patch mixing and self patch mixing.