Env-Aware Anomaly Detection

Ignore Style Changes, Stay True to Content!

Ștefan Smeu[*1,2] Elena Burceanu[*1] Andrei Liviu Nicolicioiu[3] Emanuela Haller[1]

[1] Bitdefender, Romania [2] University of Bucharest [3] MPI for Intelligent Systems, Tübingen

KEY INSIGHTS

- **Benchmark for unsupervised anomaly detection in images** - set out the differences between anomaly detection and classical (supervised) distribution shift analysis
- **Env-aware learning methods produce better embeddings** for anomaly detection
- **EA-MoCo - adjusting contrastive learning** to be aware of multiple environments improves performance

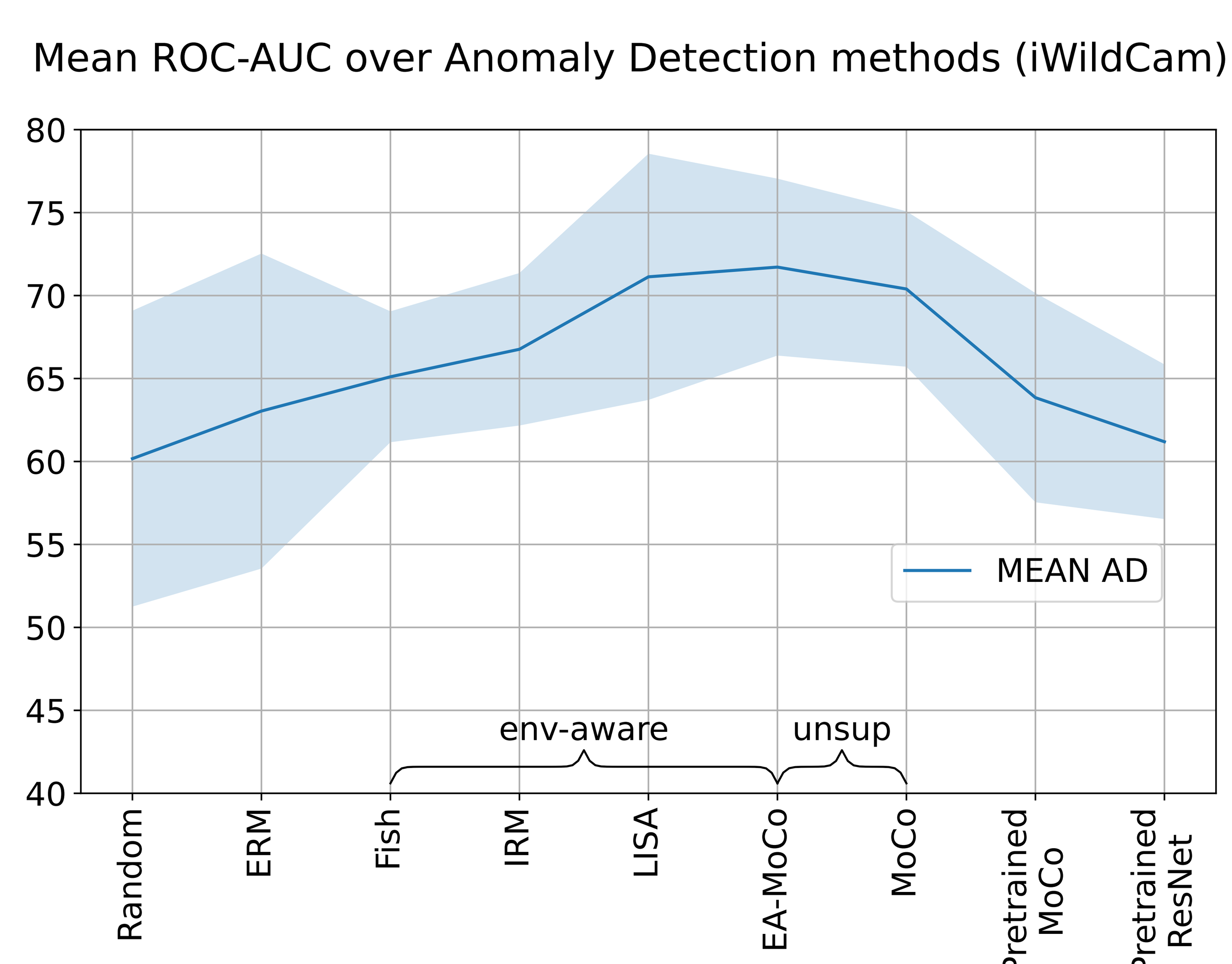
Out-of-Distribution Regimes

- We identified 4 different scenarios for changes that occur between train and test data
- Differentiate between style and content changes

	Style	Content	Description
A.	ID	ID	**Assumption:** $p_e(x_S, x_C, y)$, $p_e(x_S, x_C)$ are constant **Goal/Task:** model $p_e(y\|x)$ or $p_e(x, y)$ or $p_e(x)$ ***e.g.*** algorithms following the ERM paradigm
B.	OOD	ID	**Assumption:** $p_e(x_S)$ changes over envs - closer to real-world scenarios **Goal/Task:** same as **A.**, while being robust to Style changes ***e.g.*** IRM, V-Rex, Fish, Lisa
C.	ID	OOD	**Assumption:** $p_e(x_C)$ changes over envs **Goal/Task:** detect Content changes ***e.g.*** open set recognition; detect semantic anomalies or novelties
D.	OOD	OOD	**Assumption:** both $p_e(x_S)$, $p_e(x_C)$ change over envs - closer to real-world scenarios **Goal/Task:** same as **C.**,while being robust to Style changes ***e.g.*** **EA-MoCo** (our approach)

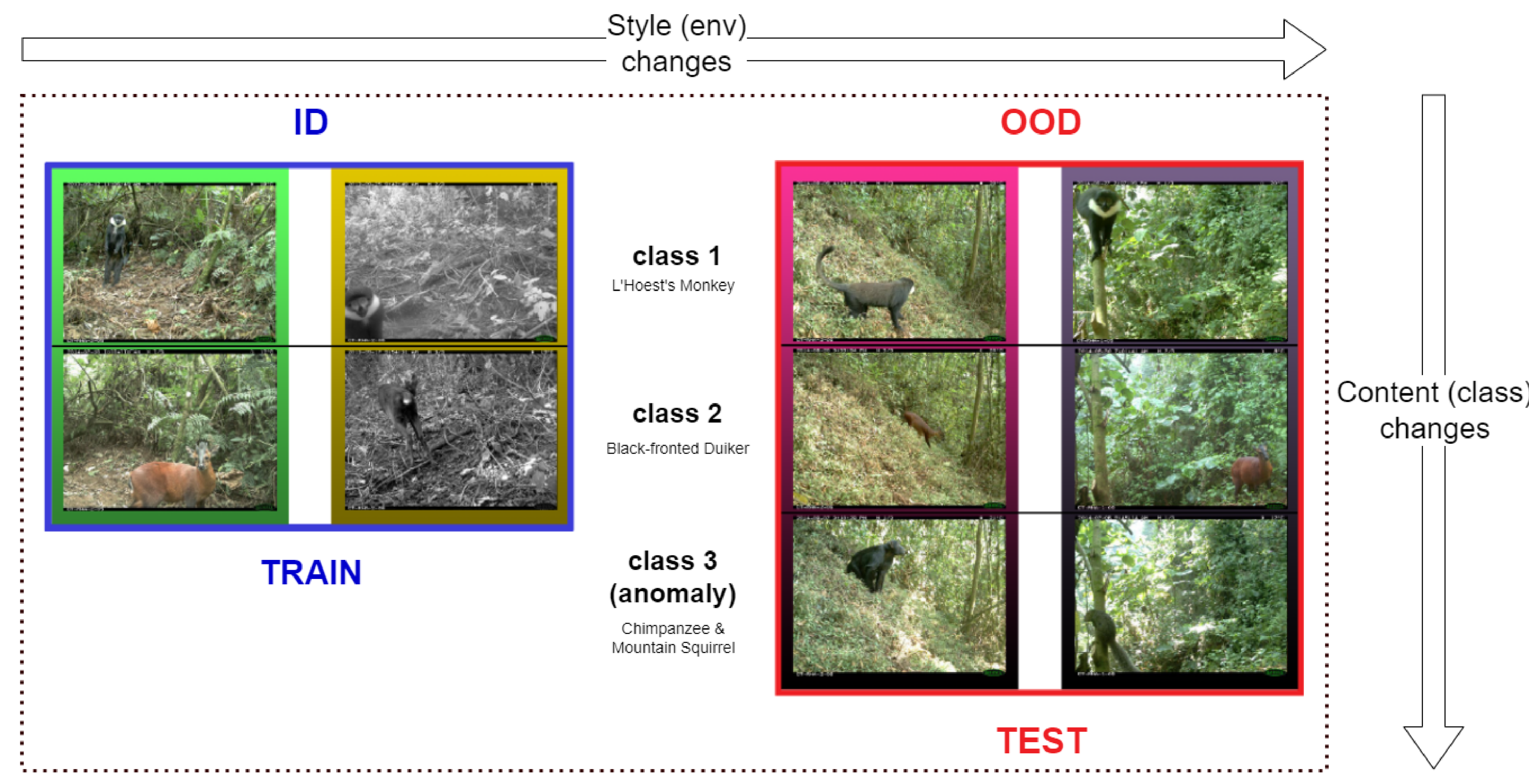
Results

	Pretrain	None	Supervised				Unsupervised		Other dataset	
		Random	ERM	Fish	IRM	Lisa	**EA-MoCo**	MoCo v3	MoCo v3	ResNet
Anom. Detect. method	IsoForest	65.2	63.1	68.0	64.3	**75.2**	70.9	68.4	64.6	61.8
	INNE	50.1	67.7	66.1	68.7	76.5	**77.0**	71.9	68.7	57.8
	LODA	65.1	63.8	66.7	66.2	**73.9**	71.1	66.9	67.1	69.9
	OCSVM	57.9	67.5	65.5	64.5	**78.4**	71.4	68.5	57.1	62.1
	PCA	64.1	40.4	63.3	64.4	55.6	**67.7**	63.9	60.9	63.2
	LOF5	43.2	61.0	59.7	61.3	65.1	60.9	**68.3**	58.5	53.2
	KNN	73.2	75.7	72.0	77.7	66.9	77.0	**78.9**	76.5	57.8
	KDE	62.6	65.1	59.4	67.0	77.4	**77.8**	76.3	57.4	63.6
	Mean AD (OOD)	60.2	63.0	65.1	66.8	71.1	**71.7**	70.4	63.8	61.2



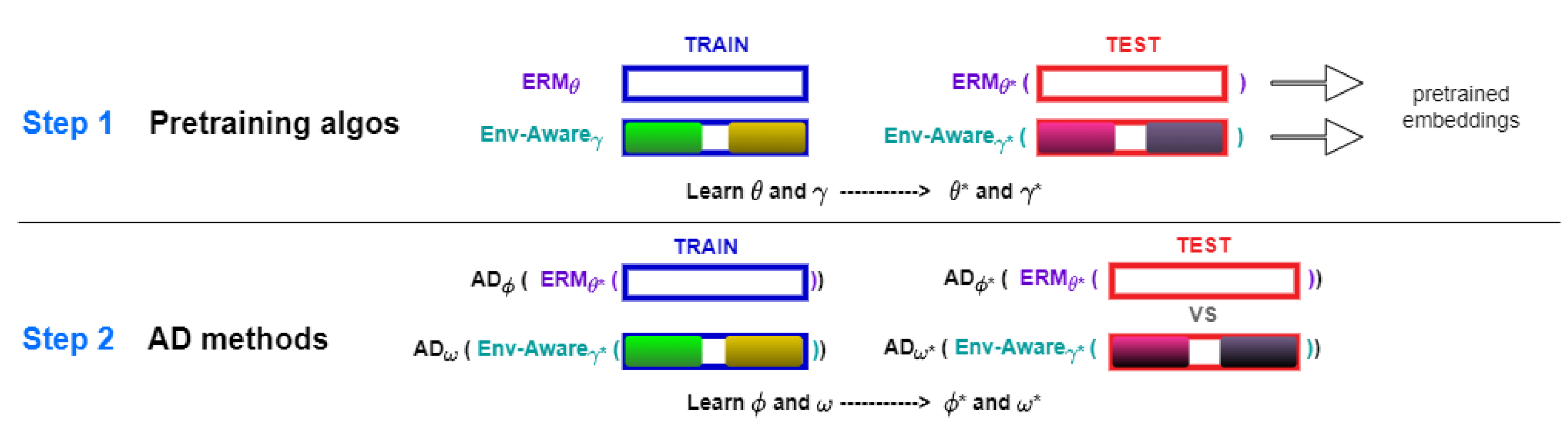
- Env-aware methods perform better
- EA-MoCo scores best on most AD methods

Anomaly Detection Setup - Style and Content OOD



Two-Step Learning Process

- Step 1 Learn embeddings robust to style changes using env-aware methods (currently, they cover only supervised tasks)
- Step 2 Anomaly detection using those learned embeddings



Data Splits

- Train data (of both steps) consists only of ID content and ID style data
- Pretraining test data consists only of OOD style, ID content data
- Anomaly detection test data consists of OOD style, ID & OOD content data

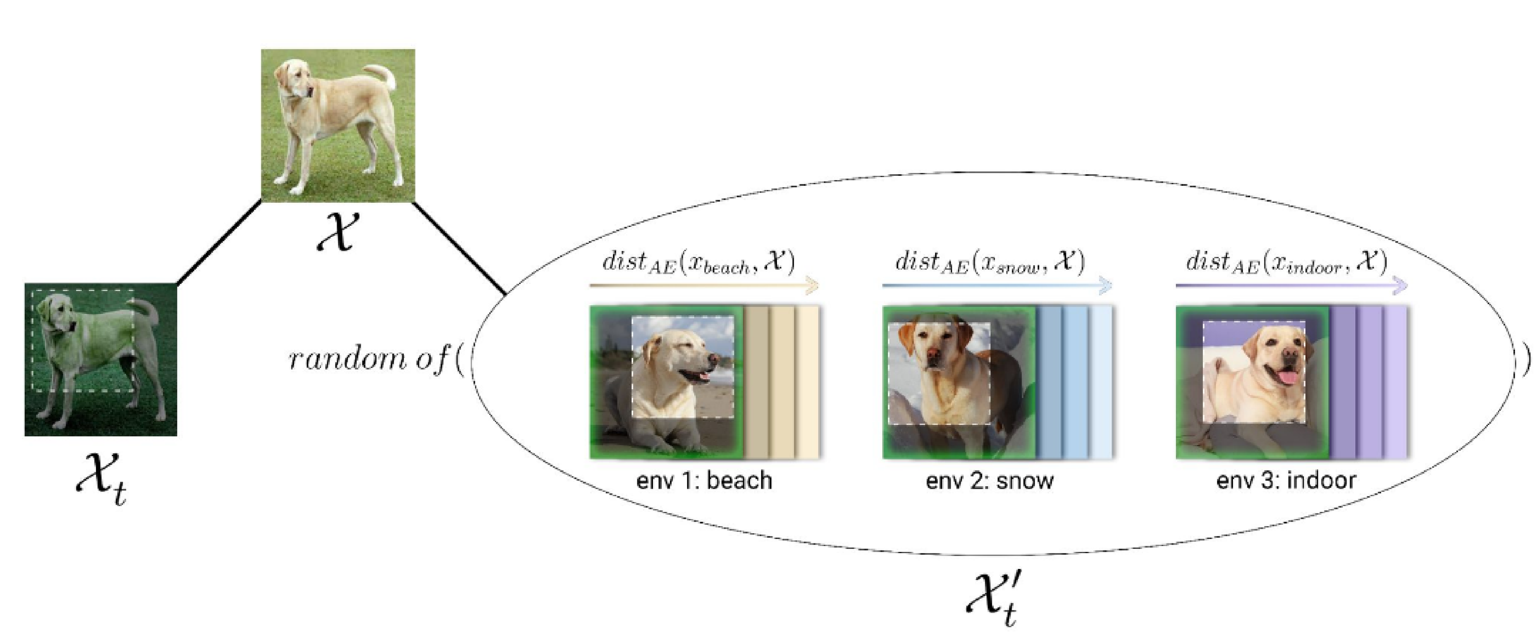
Step 1: Supervised Non-AD Pretraining

- We adapt already existing env-aware solutions for supervised learning (*e.g.* IRM, LISA, V-Rex) to anomaly detection
- We model the supervised task as a binary classification of 2 groups of labels

Step 1: Fully Unsupervised Pretraining - EA-MoCo

EA-MoCo - env-aware contrastive learning with positive pair formed of:

- usual random augmented version of anchor ($\mathcal{X}_t$)
- closest sample from a different, random environment w.r.t. trained autoencoder embeddings ($\mathcal{X}'_t$)



Contact us

{ssmeu, eburceanu}@bitdefender.com

https://bit-ml.github.io/