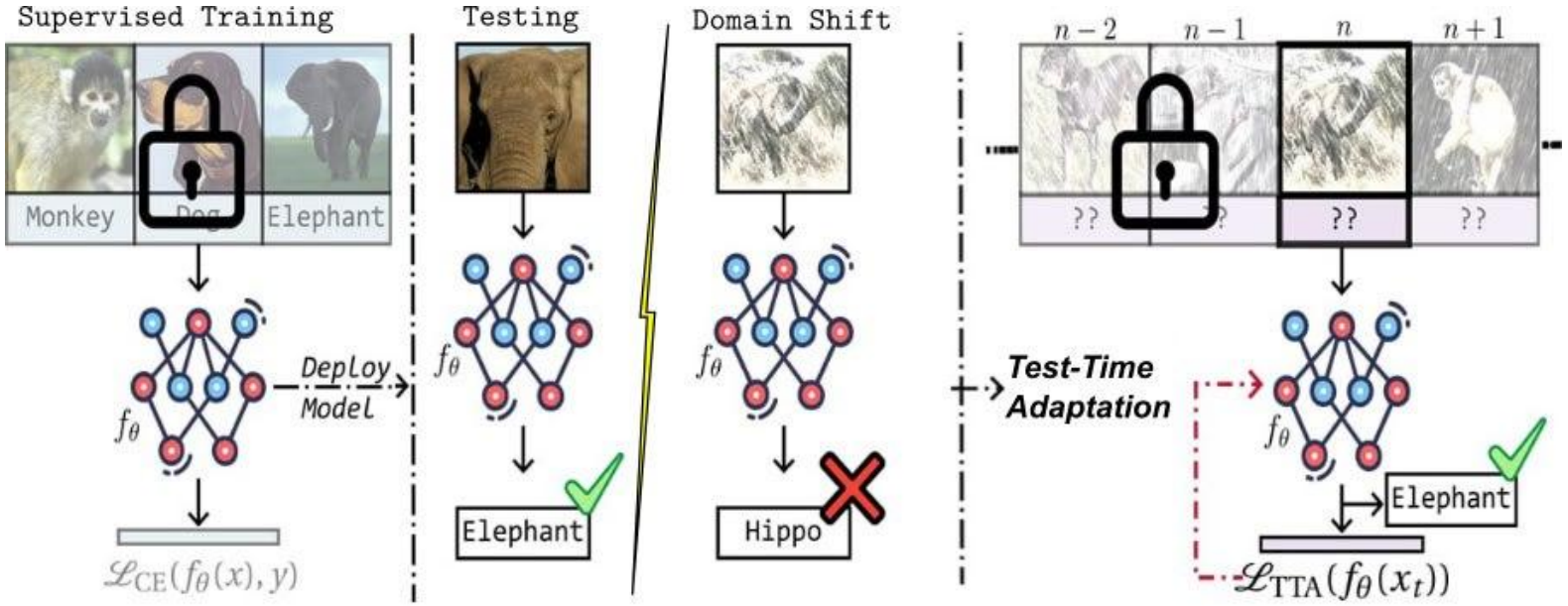
ReservoirTTA: Prolonged Test-time Adaptation for Evolving and Recurring Domains

Guillaume Vray*, Devavrat Tomar*, Xufeng Gao, Jean-Philippe Thiran, Evan Shelhamer, Behzad Bozorgtabar

Context
Test-Time Adaptation

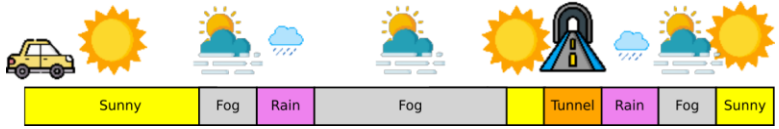
- TTA addresses domain shifts by continuously updating source model weights and making real-time predictions from incoming data streams.



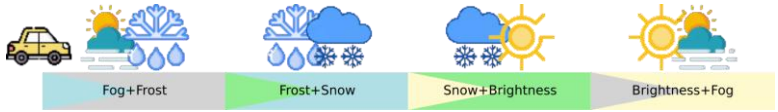
Problem
Adapting a Single Model can Fail over Longer Horizons

- **C**ontinual **S**tructural **C**hange (**CSC**)

- **C**ontinual **D**ynamic **C**hange (**CDC**)

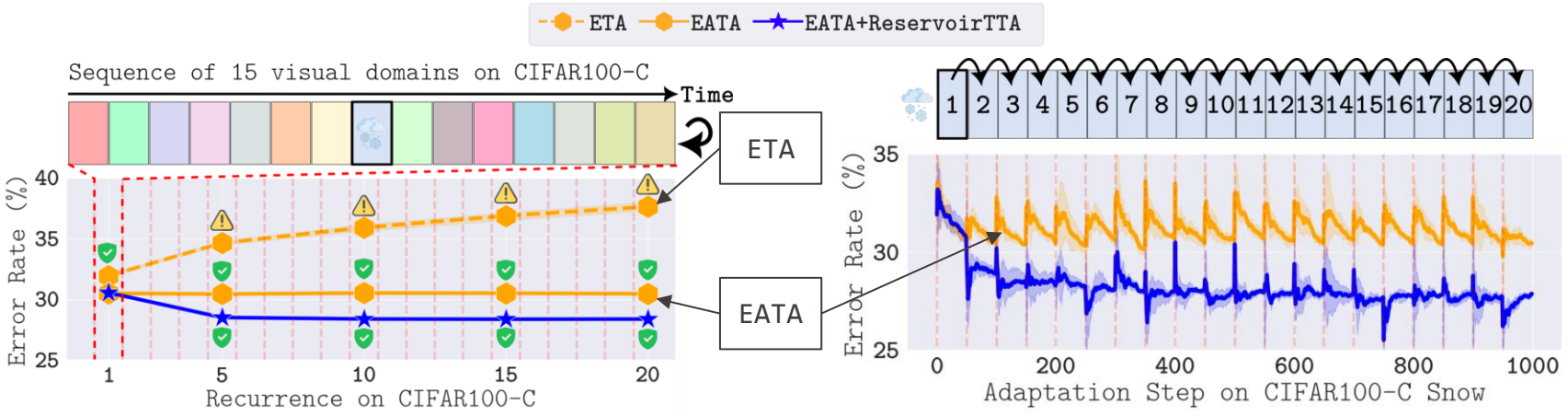


- **C**ontinuously **C**hanging **C**orruptions (**CCC**)



Problem
Adapting a Single Model can Fail over Longer Horizons

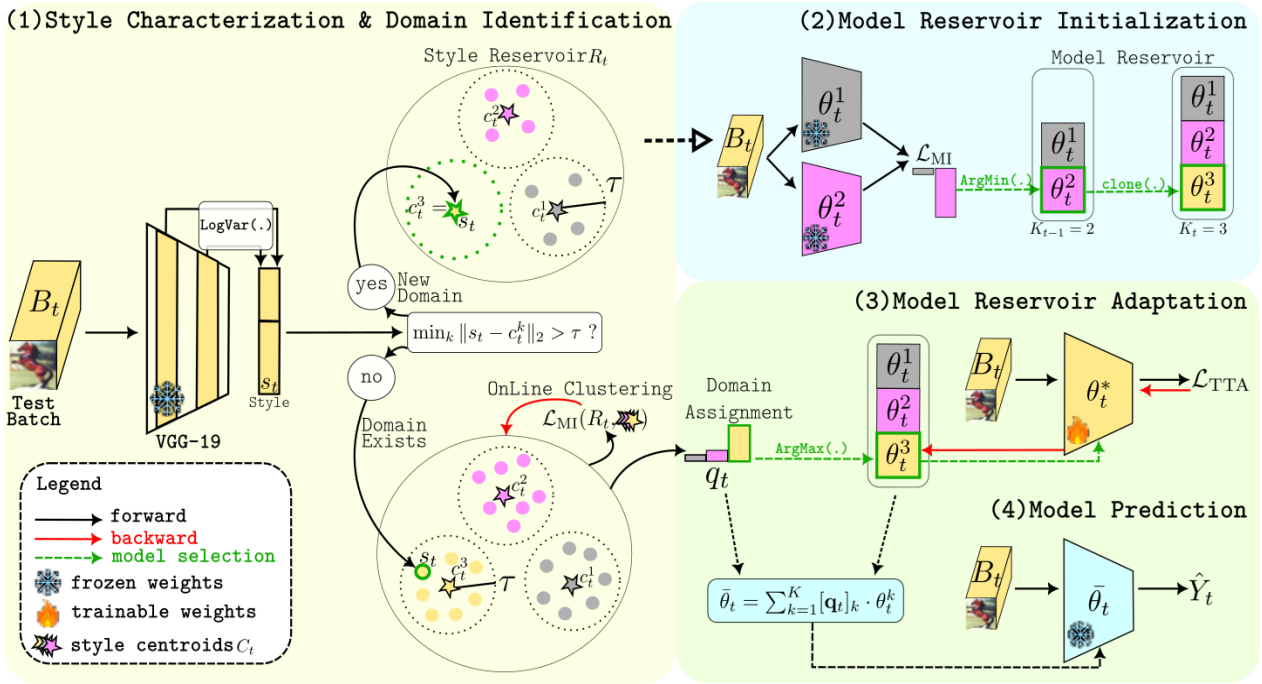
- SOTA Methods like ETA [1] degrade for prolonged test-time adaptation
  - Parameter variance growths linearly over time
- Recent works like EATA [1] are more stable
  - Weight ensembling or Fisher Regularization bounds parameter variance
- Still suffer from ***forgetting*** and lack efficient ***re-adaptation***



[1] Niu et el. (2022): **Efficient Test-Time Model Adaptation without Forgetting**

Method
Reservoir = Discovering + Updating Test-Time Experts

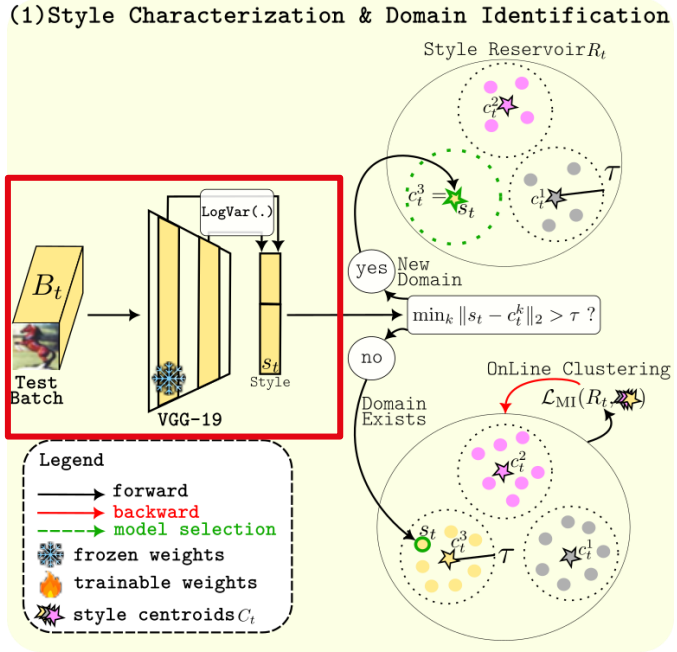
- Plug-in framework to avoid *forgetting*, promote domain specialization



Method
Reservoir = Discovering + Updating Test-Time Experts

- Statistics from VGG-19 layers to quantify **style**

$$\mathbf{s}_l(\mathbf{B}_t) = \texttt{logvar}(\mathbf{z}_l) \quad \mathbf{s}_t = \left[\mathbf{s}_1(\mathbf{B}_t), \ldots, \mathbf{s}_L(\mathbf{B}_t)\right] \in \mathbb{R}^d$$

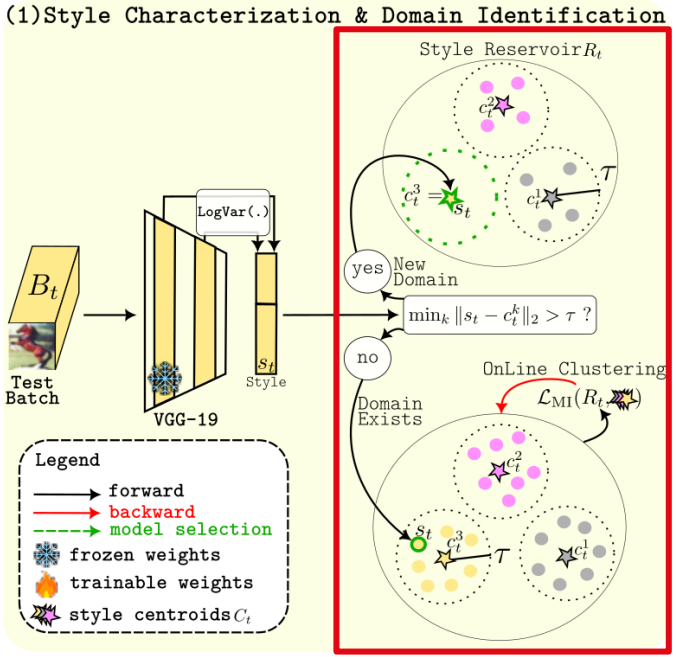


Method
Reservoir = Discovering + Updating Test-Time Experts

- Statistics from VGG-19 layers to quantify **style**

$$\mathbf{s}_l(\mathbf{B}_t) = \texttt{logvar}(\mathbf{z}_l) \quad \mathbf{s}_t = \left[\mathbf{s}_1(\mathbf{B}_t), \ldots, \mathbf{s}_L(\mathbf{B}_t)\right] \in \mathbb{R}^d$$

- Domain identification via Online Clustering

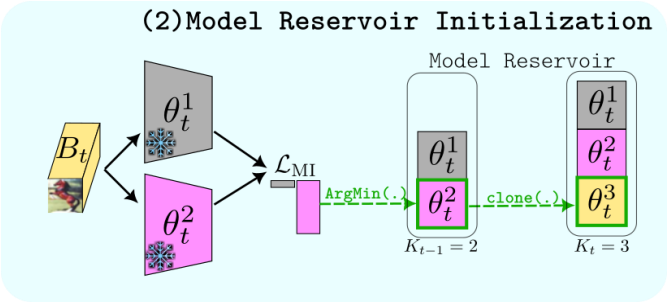


Method
Reservoir = Discovering + Updating Test-Time Experts

- Statistics from VGG-19 layers to quantify **style**

$$\mathbf{s}_l(\mathbf{B}_t) = \texttt{logvar}(\mathbf{z}_l) \quad \mathbf{s}_t = \left[\mathbf{s}_1(\mathbf{B}_t), \ldots, \mathbf{s}_L(\mathbf{B}_t)\right] \in \mathbb{R}^d$$

- Domain identification via Online Clustering
- Initialize Model Reservoir, if new domain detected

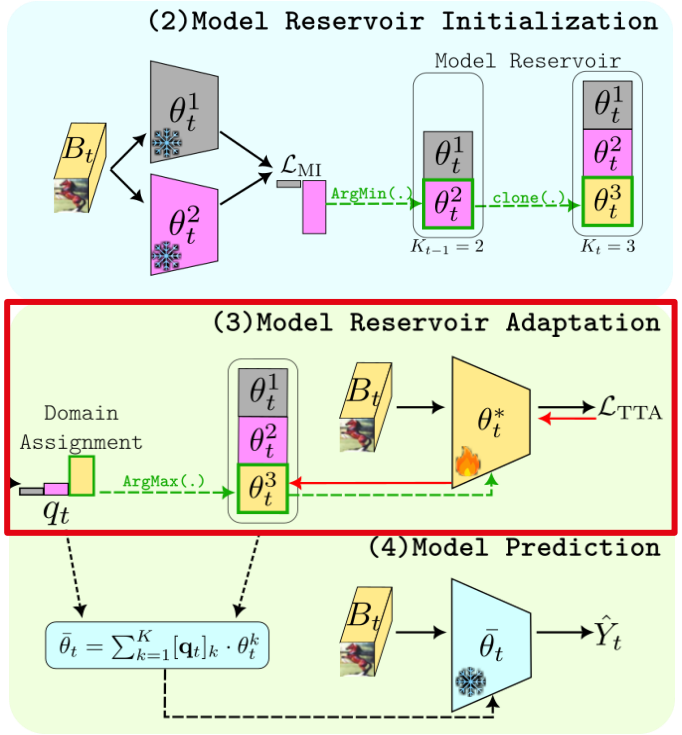


Method
Reservoir = Discovering + Updating Test-Time Experts

- Statistics from VGG-19 layers to quantify **style**

$$\mathbf{s}_l(\mathbf{B}_t) = \texttt{logvar}(\mathbf{z}_l) \quad \mathbf{s}_t = [\mathbf{s}_1(\mathbf{B}_t), \ldots, \mathbf{s}_L(\mathbf{B}_t)] \in \mathbb{R}^d$$

- Domain identification via Online Clustering
- Initialize Model Reservoir, if new domain detected
- Adapt domain specialized model



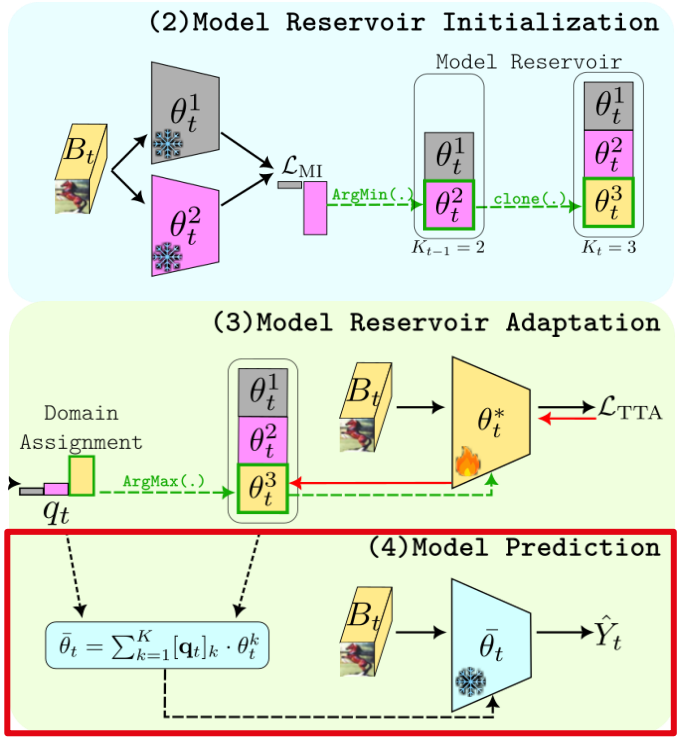
Method

Reservoir = Discovering + Updating Test-Time Experts

- Statistics from VGG-19 layers to quantify **style**

$$\mathbf{s}_l(\mathbf{B}_t) = \texttt{logvar}(\mathbf{z}_l) \quad \mathbf{s}_t = [\mathbf{s}_1(\mathbf{B}_t), \ldots, \mathbf{s}_L(\mathbf{B}_t)] \in \mathbb{R}^d$$

- Domain identification via Online Clustering
- Initialize Model Reservoir, if new domain detected
- Adapt domain specialized model
- Model ensembling for prediction



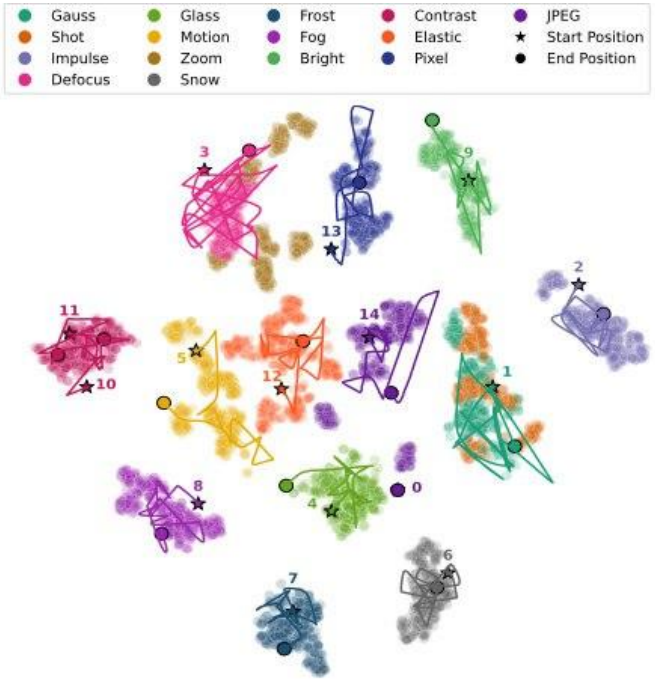
Results
Robust and Efficient Online Adaptation

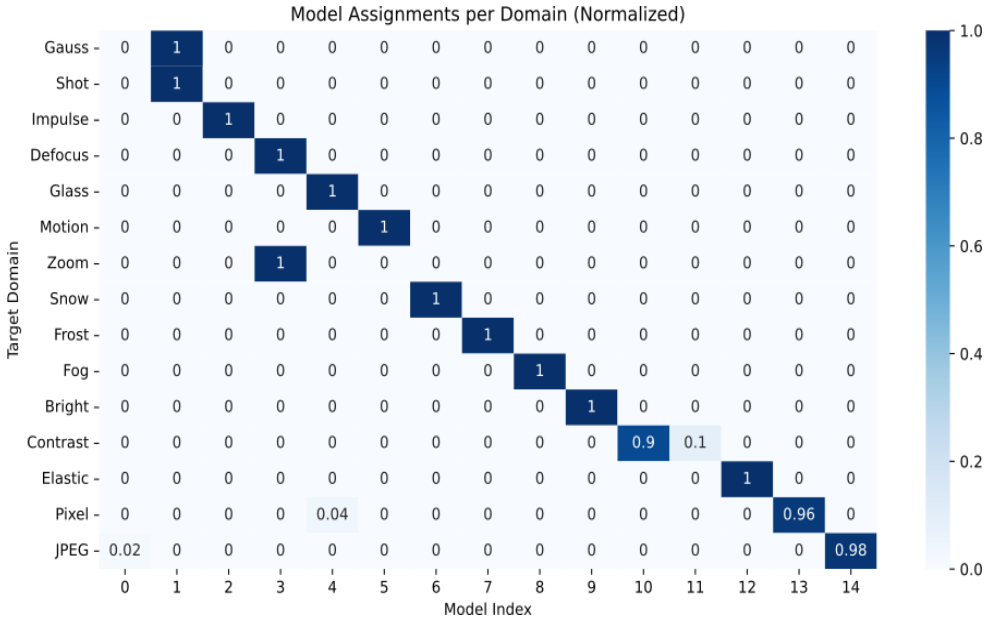
- **ReservoirTTA *consistently achieves superior performance* (Error rate %) over 20 recurring TTA visits across multiple datasets.**

	Recurring CSC									Recurring CDC								
	CIFAR-10-C			CIFAR-100-C			ImageNet-C			CIFAR-10-C			CIFAR-100-C			ImageNet-C		
	Recurring visit			*Recurring visit*			*Recurring visit*			*Recurring visit*			*Recurring visit*			*Recurring visit*		
Method	**1**	**20**	**Δ**	**1**	**20**	**Δ**	**1**	**20**	**Δ**	**1**	**20**	**Δ**	**1**	**20**	**Δ**	**1**	**20**	**Δ**
Source	43.5	43.5	+0.0	46.5	46.5	+0.0	82.0	82.0	+0.0	43.5	43.5	+0.0	46.5	46.5	+0.0	82.0	82.0	+0.0
Single-Target TTA																		
TENT (ICLR 21)	19.3	87.8	+68.5	61.4	99.0	+37.6	62.6	99.5	+36.9	20.5	87.0	+66.5	60.2	98.9	+38.7	62.0	99.5	+37.5
+ReservoirTTA	18.3	17.6	-0.7	38.1	44.0	+5.9	62.6	58.2	-4.4	18.2	17.4	-0.8	33.9	39.7	+5.8	62.4	57.5	-4.9
Continual TTA																		
CoTTA* (CVPR 22)	18.8	22.4	+3.6	35.1	65.5	+30.4	67.6	62.7	-4.9	18.8	22.3	+3.5	35.1	65.1	+30.0	67.7	61.5	-6.2
RoTTA (CVPR 23)	19.4	18.4	-1.0	34.8	59.1	+24.3	67.3	99.4	+32.1	21.9	20.4	-1.5	36.8	73.8	+37.0	71.6	99.5	+27.9
ETA (ICML 22)	17.8	30.9	+13.1	32.0	37.6	+5.6	60.0	59.4	-0.6	17.9	33.5	+15.6	32.4	37.6	+5.2	59.3	60.1	+0.8
+ReservoirTTA	17.5	**16.4**	-1.1	31.6	30.0	-1.6	59.8	53.1	-6.7	**17.4**	**16.3**	-1.1	30.9	29.7	-1.2	58.6	52.2	-6.4
SAR (ICLR 23)	20.4	20.4	+0.0	31.9	60.4	+28.5	61.9	67.1	+5.2	20.4	20.4	+0.0	31.6	57.8	+26.2	61.5	66.2	+4.7
+ReservoirTTA	20.4	20.4	+0.0	31.9	30.5	-1.4	62.2	53.1	-9.1	20.4	20.4	+0.0	31.7	29.8	-1.9	62.6	53.6	-9.0
Persistent TTA																		
RDumb (NeurIPS 23)	17.8	18.4	+0.6	32.0	32.9	+0.9	59.8	56.8	-3.0	17.9	18.1	+0.2	32.4	32.6	+0.2	59.6	59.5	-0.1
PeTTA (NeurIPS 24)	23.0	17.2	-5.8	39.4	32.9	-6.5	67.5	60.1	-7.4	27.2	20.8	-6.4	42.1	35.3	-6.8	71.6	69.5	-2.1
EATA (ICML 22)	**17.5**	17.8	+0.3	30.5	30.5	+0.0	57.5	55.9	-1.6	17.7	17.9	+0.2	31.0	31.1	+0.1	58.5	57.0	-1.5
+ReservoirTTA	17.5	**16.4**	-1.1	30.6	28.4	-2.2	58.0	**51.0**	-7.0	17.5	16.4	-1.1	30.4	28.4	-2.0	58.5	**51.8**	-6.7
ROID* (WACV 24)	17.8	17.7	-0.1	**29.5**	29.3	-0.2	**56.1**	55.5	-0.6	18.0	18.1	+0.1	30.2	30.1	-0.1	58.7	58.3	-0.4
+ReservoirTTA	17.8	16.8	-1.0	29.6	**27.8**	-1.8	56.4	52.1	-4.3	17.9	16.8	-1.1	**29.6**	**27.8**	-1.8	**57.0**	53.0	-4.0

Results
Robust Model Assignment

- Style features ~ Domain





Model assignments per domain

Summary and Conclusion

- ReservoirTTA extends test-time adaptation to multiple models, dynamically updating domain specialists instead of forcing a single model to adapt continuously.
- Robustness under domain shifts is achieved through dynamic clustering of style features, reducing catastrophic forgetting and stabilizing adaptation
- Trade-off: the reservoir design adds ~30% computational overhead and relies on per-batch updates, but memory cost is low and adaptive update triggers could improve efficiency.

Code

Thank you

Paper