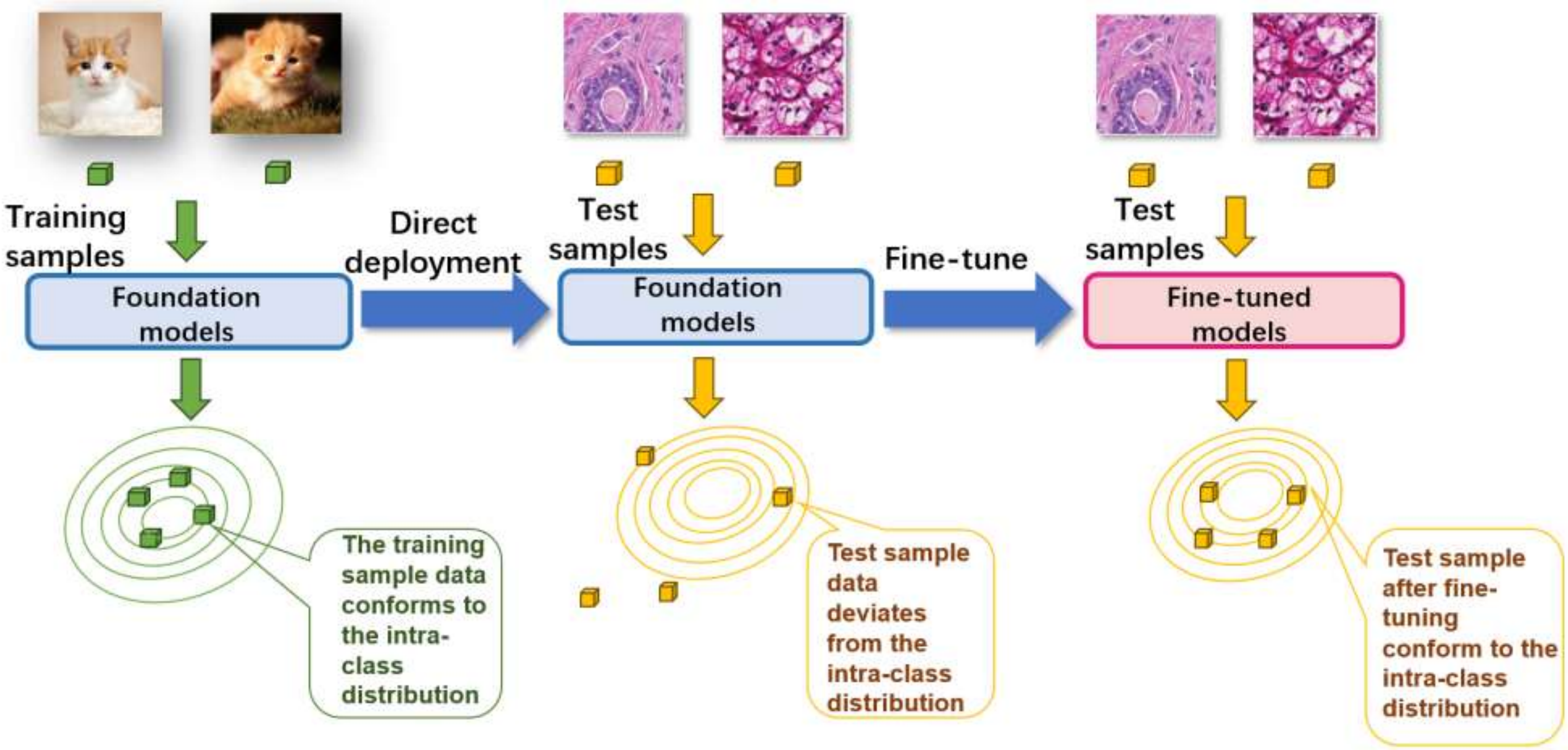
DOTA: **D**istributi**O**nal **T**est-time **A**daptation of Vision-Language Models

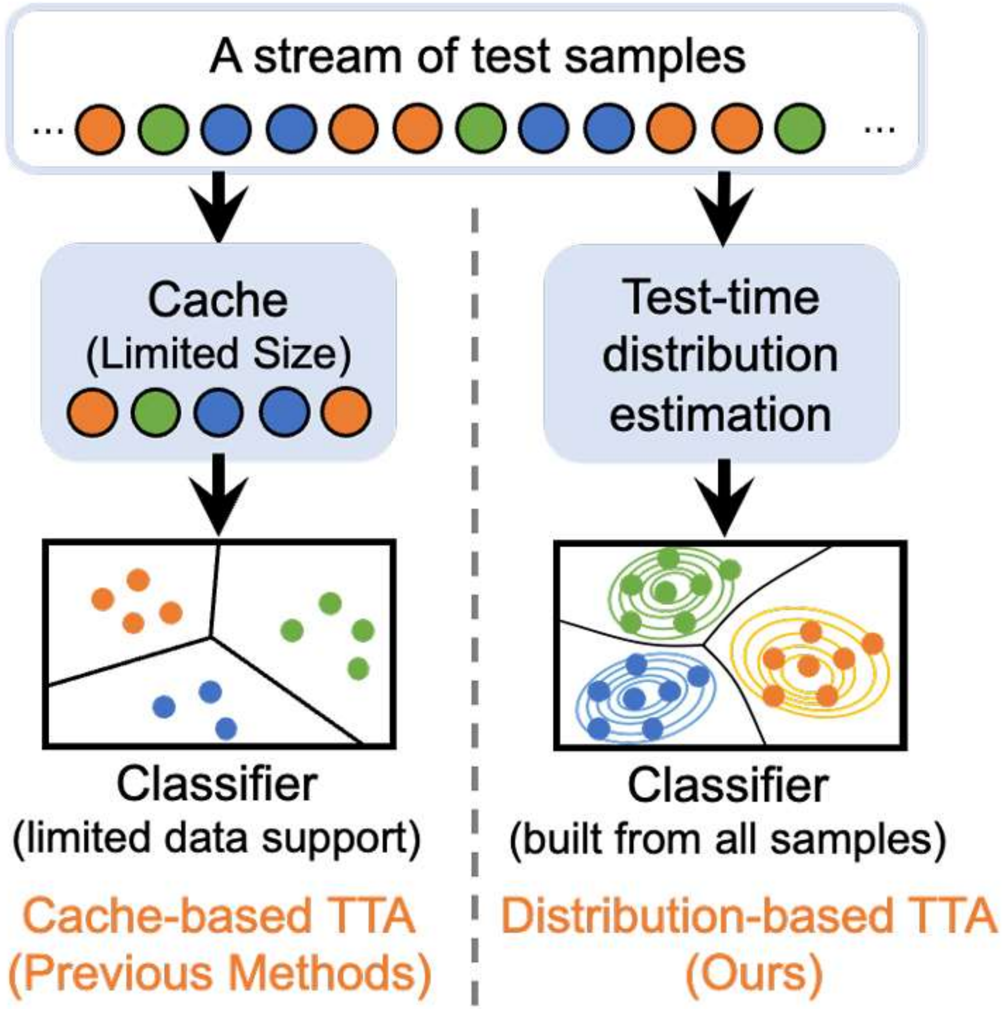
Zongbo Han*, Jialong Yang, Guangyu Wang, Junfan Li, Qianli Xu, Mike Zheng Shou*,Changqing Zhang*

Introduction：Test-Time Adaptation



Test-Time Adaptation (TTA) aims to enhance the model's capability to handle downstream tasks during deployment, without requiring access to test data labels.

Advantages Compared to Cache-based Methods



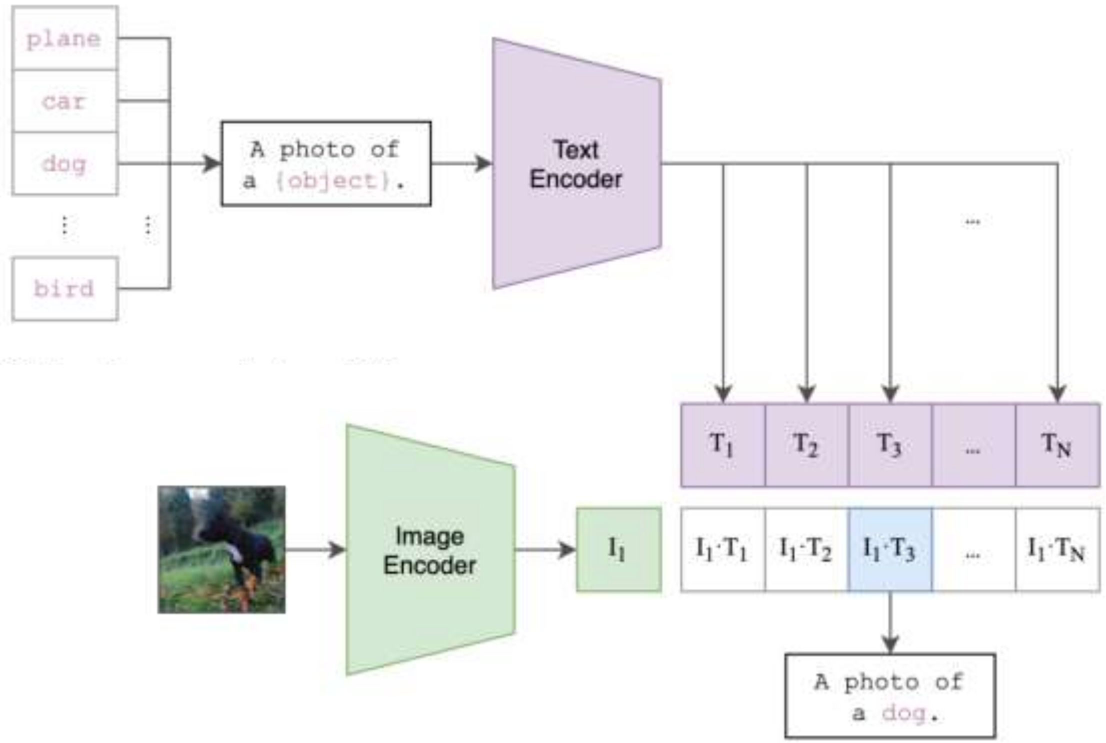
- Cache-based TTA methods[1,2] store individual test samples within a **limited cache**, which often leads to **underutilization of the available test data.**
- In contrast, DOTA continuously estimates the **underlying distribution** of the test data, enabling the **full exploitation of all available test samples.**

[1] Efficient test-time adaptation of vision-language models.
[2] Advancing Reliable Test-Time Adaptation of Vision-Language Models under Visual Variations.

Zero-Shot Classification with Prompt

CLIP can perform **zero-shot classification without additional training**. It applies softmax over cosine similarities between input x and class prompt weights w_k, scaled by temperature τ.

$$P_k^{\text{zs}}(y = k|\boldsymbol{x}) = \frac{\exp(\cos(\boldsymbol{x}, \boldsymbol{w}_k)/\tau)}{\sum_{k=1}^{K} \exp(\cos(\boldsymbol{x}, \boldsymbol{w}_k)/\tau)}$$



Classification with classical Gaussian discriminant analysis(GDA)

1. We assume that the embedding distribution of each class k follows a Gaussian distribution

$$P(\boldsymbol{x}|y{=}k) = \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

where μ_k and Σ_k are the mean vector and covariance matrix of class k, respectively.

2. Using Bayes' theorem, the posterior probability P(y=k|x) of class k can be given by

$$P(y{=}k|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y{=}k)P(y{=}k)}{P(\boldsymbol{x})}$$

$$P(\boldsymbol{x}) = \sum_{k=1}^{K} P(\boldsymbol{x}|y{=}k)P(y{=}k)$$

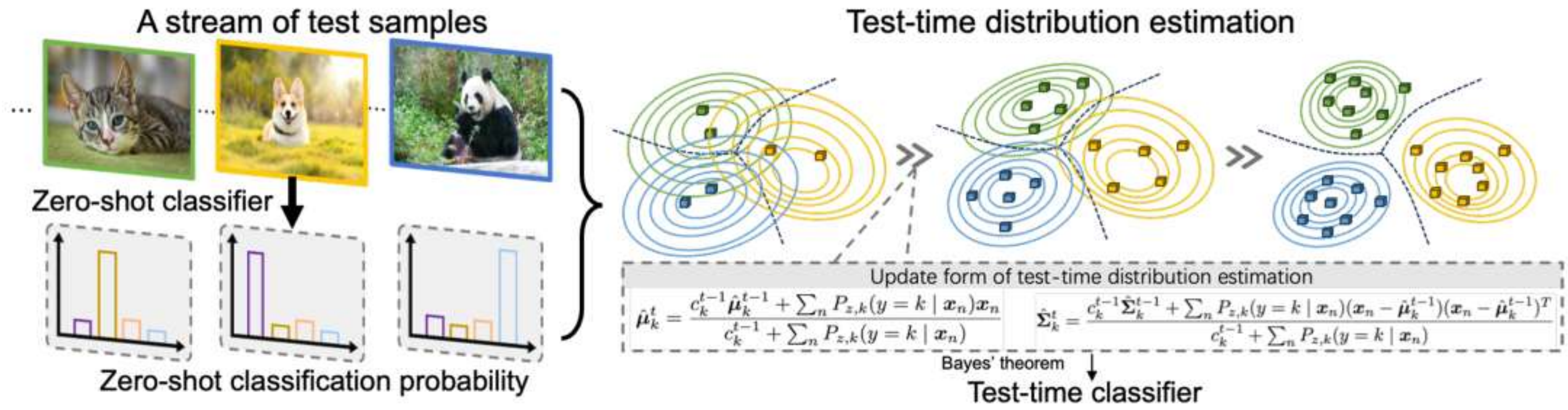
$$P(y=k)=1/k$$

$$P(y{=}\,k\,|\boldsymbol{x}) = \frac{\exp(f_k(\boldsymbol{x}))}{\sum_{k=1}^{K}\exp(f_k(\boldsymbol{x}))}$$

$$f_k(\boldsymbol{x}) = -\tfrac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k) - \tfrac{1}{2}\log|\boldsymbol{\Sigma}_k|$$

The discriminant function fk(x) measures how well a sample x fits the distribution of class k.

DistributiOnal Test-time Adaptation(DOTA)



DistributiOnal Test-time Adaptation(DOTA) Process Framework

Update of $\{\hat{\mu}_k, \hat{\Sigma}_k\}_{k=1}^K$

$$\hat{\mu}_k^t = \frac{c_k^{t-1}\hat{\mu}_k^{t-1} + \sum P_k^{\text{zs}}(y=k \mid x_n)x_n}{c_k^{t-1} + \sum P_k^{\text{zs}}(y=k \mid x_n)}, \hat{\Sigma}_k^t = \frac{c_k^{t-1}\hat{\Sigma}_k^{t-1} + \sum P_k^{\text{zs}}(y=k \mid x_n)S_k^{t-1}}{c_k^{t-1} + \sum P_k^{\text{zs}}(y=k \mid x_n)}$$

c_k^{t-1} represents the effective sample size, defined by the cumulative confidences of the observed samples of class k at step $t-1$

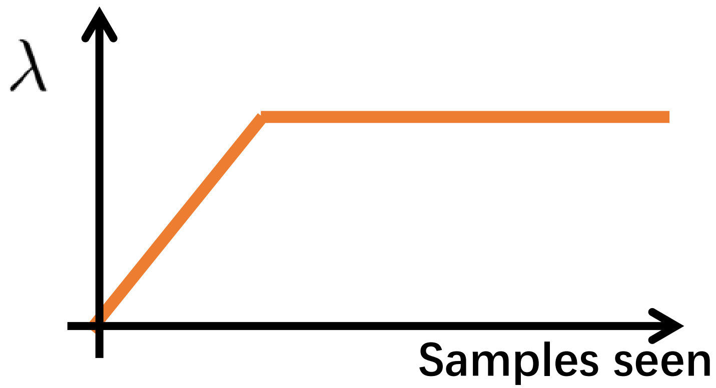
Adaptive fusion of zero-shot and test-time classifier

zero-shot classification and test-time result fusion approach：

$$P_k(y=k|x) = \frac{\exp(\cos(\boldsymbol{x}, \boldsymbol{w}_k)/\tau + \lambda f_k(\boldsymbol{x}))}{\sum_{k=1}^{K}[\exp(\underbrace{\cos(\boldsymbol{x}, \boldsymbol{w}_k)}_{CLIP\_logit}/\tau + \lambda \underbrace{f_k(\boldsymbol{x})}_{DOTA\_logit})]}$$

$$\lambda = \min(\rho c, \eta)$$

This approach encourages the model to rely on the zero-shot classifier results when the test samples are insufficient to estimate the distribution, mitigating the potential negative impact of the test-time classifier.



Main Results

Top-1 accuracy(%) under the cross-domain generalization scenario

Method	Aircraft	Caltech101	Cars	DTD	EuroSAT	Flower102	Food101	Pets	SUN397	UCF101	Average
Zero-Shot	23.22	93.55	66.11	45.04	50.42	66.99	82.86	86.92	65.63	65.16	64.59
TPT	24.78	94.16	66.87	47.75	42.44	68.98	84.67	87.79	65.50	68.04	65.10
DiffTPT	25.60	92.49	67.01	47.00	43.13	70.10	87.23	88.22	65.74	62.67	65.47
TDA	23.91	94.24	67.28	47.40	58.00	71.42	86.14	88.63	67.62	70.66	67.53
BoostAdapter	27.45	94.77	69.30	45.69	61.22	71.66	87.17	89.51	68.09	71.93	68.68
HisTPT	26.90	94.50	69.20	48.90	49.70	71.20	89.30	89.10	67.20	70.10	67.60
ZERO	25.21	93.66	68.04	46.12	34.33	67.68	86.53	87.75	65.03	67.77	64.21
Dota	26.25	94.16	69.56	47.64	62.78	75.23	87.08	92.01	69.80	72.54	69.71
DMN w/PE	30.03	95.38	67.96	55.85	59.43	74.49	85.08	92.04	70.18	72.51	70.30
Dota w/ PE	29.82	94.85	69.06	55.97	58.35	77.06	87.07	92.40	70.97	74.86	71.04

Top-1 accuracy(%) under the cross-domain generalization under the NDS scenario

Method	ImageNet	ImageNet-A	ImageNet-R	ImageNet-S	Average
Zero-Shot	68.34	49.89	77.65	48.24	61.03
TPT	68.98	54.77	77.06	47.94	62.19
DiffTPT	70.30	55.68	75.00	46.80	61.95
TDA	69.51	60.11	80.24	50.54	65.10
ZERO	69.31	59.61	77.22	48.40	63.64
Dota	70.69	61.50	81.21	51.84	66.31

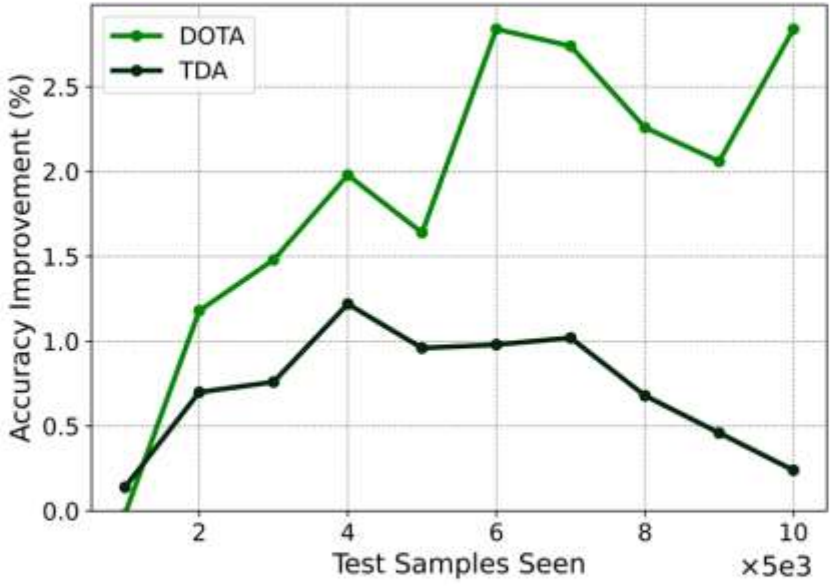
Insights of DOTA

Efficiency and effectiveness

Method	Testing Time	Accuracy	Gain
Zero-Shot	11.82min	68.34	0
TPT	447min	68.98	+0.64
DiffTPT	1346min	70.30	+1.96
TDA	**22min**	69.51	+1.17
Dota (Ours)	**22min**	**70.69**	**+2.35**

- DOTA is **faster** than the methods that require gradient back propagation
- compared with TDA, the speed of DOTA is comparable, but the **performance is higher**

Ability of continuous learning



- DOTA **progressively enhances model performance** as the number of test samples increases
- TDA shows an initial improvement that **subsequently declines**.

Insights of DOTA

Necessity of distribution estimation

Method	Aircraft	Caltech101	Cars	DTD	EuroSAT
Dota	26.25	94.16	69.56	47.64	62.78
w/o covariance	25.29 -0.96	94.16 0.00	67.47 -2.09	45.62 -2.02	55.06 -7.72

Method	Flower102	Food101	Pets	SUN397	UCF101
Dota	75.23	87.08	92.01	69.80	72.54
w/o covariance	71.34 -3.89	86.44 -0.64	90.57 -1.44	67.88 -1.92	69.34 -3.20

- **Design:** Compare full DOTA (updates mean + covariance) vs. simplified version (updates only mean, no covariance).
- **Results: Accuracy consistently drops** without covariance updates

Robust to hyperparameters

σ^2	0.0001	0.001	0.002	0.004	0.008	0.02
Acc	70.72	70.72	70.69	70.70	70.60	70.42

$\eta\backslash\rho$	0.005	0.01	0.02	0.03
0.2	70.69	70.66	70.59	70.54
0.3	70.64	70.55	70.36	70.28
0.4	70.64	70.51	70.24	70.13
0.5	70.64	70.48	70.15	70.00

- all hyperparameter combinations show that the proposed method outperforms the original zero-shot classifier

Thank you!