
Towards Backpropagation-Free and Distribution-Aware Test-Time Adaptation

Anonymous Author(s)

Affiliation

Address

email

1 This appendix provides a detailed theoretical analysis of our method, along with additional experi-
2 mental results. The contents are organized as follows:

3 • **Appendix A: Theoretical Analysis**

- 4 A.1 Proof of Gaussian Discriminant Analysis
- 5 A.2 Online Test-time Distribution Estimation
- 6 A.3 Transductive Test-time Distribution Estimation
- 7 A.4 Limitations and Discussion

8 • **Appendix B: Additional Experimental Results**

- 9 B.1 Experiments with CLIP-RN50 Backbone
- 10 B.2 Few-shot Adaptation
- 11 B.3 Evaluation with Different VLMs
- 12 B.4 Additional Analysis

13 **A Theoretical Analysis**

14 **A.1 Proof of Gaussian Discriminant Analysis**

15 This subsection provides the Gaussian Discriminant Analysis (GDA) decision rule under class-
16 conditional Gaussian assumptions and serves as the theoretical justification for Eq. (4) in the main
17 paper. Specifically, we derive the discriminant function of GDA under the assumption that class-
18 conditional distributions follow multivariate Gaussians with a shared covariance matrix:

$$\mathbb{P}(\mathbf{x}|y_k) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp \left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right). \quad (1)$$

19 Using Bayes' rule, the posterior probability can be expressed as: $\mathbb{P}(y_k|\mathbf{x}) = \frac{\mathbb{P}(\mathbf{x}|y_k) \mathbb{P}(y_k)}{\mathbb{P}(\mathbf{x})} \propto$
20 $\pi_k \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma})$, which reformulates the K -way classification problem into a maximum likelihood
21 estimation task. Considering that the target source domain generally is class-balanced, we set the
22 prior class distribution to be uniform: $\mathbb{P}(y_k) = \frac{1}{K}$. Then, we can derive the GDA classifier by

23 maximizing the likelihood as following:

$$\log \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}) \cdot \pi_k = \log \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp \left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right) + \log \pi_k \quad (2)$$

$$= -\frac{d}{2} \log 2\pi - \frac{1}{2} \log |\boldsymbol{\Sigma}| - \frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) + \log \pi_k \quad (3)$$

$$= -\frac{d}{2} \log 2\pi - \frac{1}{2} \log |\boldsymbol{\Sigma}| - \frac{1}{2} \mathbf{x}^\top \boldsymbol{\Sigma}^{-1} \mathbf{x} + \boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1} \mathbf{x} \quad (4)$$

$$- \frac{1}{2} \boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k + \log \pi_k \quad (5)$$

$$= \boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1} \mathbf{x} - \frac{1}{2} \boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k + \text{C} \quad (6)$$

$$\stackrel{(a)}{=} \boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1} \mathbf{x} - \frac{1}{2} \boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k \quad (7)$$

$$= \mathbf{w}_k^\top \mathbf{x} + b_k. \quad (8)$$

24 $\stackrel{(a)}{=}$ holds as we can remove all constant terms $\text{C} = \log \pi_k - \frac{d}{2} \log 2\pi - \frac{1}{2} \log |\boldsymbol{\Sigma}| - \frac{1}{2} \mathbf{x}^\top \boldsymbol{\Sigma}^{-1} \mathbf{x}$.
 25 Then, the GDA prediction can be expressed as: $\tilde{y}_{i,k} = \mathbf{w}_k^\top \mathbf{x}_i + b_k$, where $\mathbf{w}_k = \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k$, $b_k =$
 26 $-\frac{1}{2} \boldsymbol{\mu}_k^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k$.

27 A.2 Online Test-time Distribution Estimation

28 While GDA offers a principled framework for label estimation using class-conditional statistics, its
 29 direct application in the online setting may result in biased likelihood estimates particularly in early
 30 stages where data is scarce and predictions tend to be overconfident. To address this, we construct
 31 a regularized objective (Eq. (5) in our main paper) that integrates three complementary sources of
 32 information: (i) the online negative log-likelihood, $-z_i^\top \log \mathbb{P}_i$; (ii) a prior regularization term based
 33 on CLIP zero-shot predictions, $\mathcal{R}(z_i; \hat{y}_i)$; and (iii) a consistency regularization term guided by the
 34 knowledge bank, $\mathcal{R}(z_i; \mathcal{B})$. We formulate the full objective as follows:

$$\mathcal{L}_{\text{online}}(z_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = -z_i^\top \log \mathbb{P}_i + \mathcal{R}(z_i; \hat{y}_i) + \mathcal{R}(z_i; \mathcal{B}) \quad (9)$$

$$= -z_i^\top \log \mathbb{P}_i + \text{KL}(z_i \| \hat{y}_i) + \beta \sum_{k=1}^K \text{KL} \left(\mathcal{N}(\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k) \| \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}) \right) \quad (10)$$

$$- \sum_{j \in \mathcal{B}} \hat{y}_j^\top \log \mathbb{P}_j - \sum_{j \in \mathcal{B}} w_{ij} z_i^\top \hat{y}_j. \quad (11)$$

35 For the third term, we assume that x follows a Gaussian distribution $\mathcal{N}_0 = \mathcal{N}(\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k)$. According
 36 Matrix Cookbook [15] and [25], the following identity holds: $\mathbb{E}_{\mathcal{N}_0} [(x - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1} (x - \boldsymbol{\mu}_k)] =$
 37 $(\hat{\boldsymbol{\mu}}_k - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1} (\hat{\boldsymbol{\mu}}_k - \boldsymbol{\mu}_k) + \text{Tr}(\boldsymbol{\Sigma}^{-1} \hat{\boldsymbol{\Sigma}}_k)$. Based on this, the KL divergence between the distributions
 38 $\mathcal{N}_0 = \mathcal{N}(\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k)$ and $\mathcal{N}_1 = \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$ is given as:

$$\text{KL}(\mathcal{N}(\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k) \| \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})) = \int_x \mathcal{N}_0(x) \log \frac{\mathcal{N}_0(x)}{\mathcal{N}_1(x)} dx = \mathbb{E}_{\mathcal{N}_0} [\log \mathcal{N}_0(x) - \log \mathcal{N}_1(x)] \quad (12)$$

$$= \mathbb{E}_{\mathcal{N}_0} \left[\frac{1}{2} \log \frac{|\boldsymbol{\Sigma}|}{|\hat{\boldsymbol{\Sigma}}_k|} - \frac{1}{2} (x - \hat{\boldsymbol{\mu}}_k)^\top \hat{\boldsymbol{\Sigma}}_k^{-1} (x - \hat{\boldsymbol{\mu}}_k) + \frac{1}{2} (x - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1} (x - \boldsymbol{\mu}_k) \right] \quad (13)$$

$$= \frac{1}{2} \left(\log \frac{|\boldsymbol{\Sigma}|}{|\hat{\boldsymbol{\Sigma}}_k|} - \mathbb{E}_{\mathcal{N}_0} \left[(x - \hat{\boldsymbol{\mu}}_k)^\top \hat{\boldsymbol{\Sigma}}_k^{-1} (x - \hat{\boldsymbol{\mu}}_k) \right] + \mathbb{E}_{\mathcal{N}_0} \left[(x - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1} (x - \boldsymbol{\mu}_k) \right] \right) \quad (14)$$

$$\stackrel{(b)}{=} \frac{1}{2} \left(\log \frac{|\boldsymbol{\Sigma}|}{|\hat{\boldsymbol{\Sigma}}_k|} - d + (\hat{\boldsymbol{\mu}}_k - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1} (\hat{\boldsymbol{\mu}}_k - \boldsymbol{\mu}_k) + \text{Tr}(\boldsymbol{\Sigma}^{-1} \hat{\boldsymbol{\Sigma}}_k) \right), \quad (15)$$

39 where d is the feature dimension and $\stackrel{(b)}{=}$ holds as:

$$\mathbb{E}_{\mathcal{N}_0} \left[(x - \hat{\mu}_k)^\top \hat{\Sigma}_k^{-1} (x - \hat{\mu}_k) \right] = \mathbb{E}_{\mathcal{N}_0} \left[\text{Tr} \left((x - \hat{\mu}_k)(x - \hat{\mu}_k)^\top \hat{\Sigma}_k^{-1} \right) \right] \quad (16)$$

$$= \text{Tr}(\mathbb{E}_{\mathcal{N}_0} [(x - \hat{\mu}_k)(x - \hat{\mu}_k)^\top] \hat{\Sigma}_k^{-1}) \quad (17)$$

$$= \text{Tr}(\hat{\Sigma}_k \hat{\Sigma}_k^{-1}) \quad (18)$$

$$= \text{Tr}(\mathbb{I}_K) \quad (19)$$

$$= d \quad (20)$$

40 A.2.1 Estimation of Class Mean

41 To enable efficient distribution estimation in streaming scenarios where test samples arrive sequentially, we estimate the class means μ_k by taking the derivative of $\mathcal{L}_{\text{online}}$ w.r.t. μ_k :

$$\frac{\partial \mathcal{L}_{\text{online}}}{\partial \mu_k} = -\frac{\partial}{\partial \mu_k} (z_i^\top \log \mathbb{P}_i) + \frac{\partial}{\partial \mu_k} (\beta \sum_{k=1}^K \text{KL}(\mathcal{N}(\hat{\mu}_k, \hat{\Sigma}_k) \parallel \mathcal{N}(\mu_k, \Sigma))) \quad (21)$$

$$- \frac{\partial}{\partial \mu_k} (\sum_{j \in \mathcal{B}} \hat{y}_j^\top \log \mathbb{P}_j) \quad (22)$$

$$= -\frac{\partial}{\partial \mu_k} (z_i \log(\frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp(-\frac{1}{2}(\mathbf{x}_i - \mu_k)^\top \Sigma^{-1}(\mathbf{x}_i - \mu_k)))) \quad (23)$$

$$+ \frac{\partial}{\partial \mu_k} (\beta \text{KL}(\mathcal{N}(\hat{\mu}_k, \hat{\Sigma}_k) \parallel \mathcal{N}(\mu_k, \Sigma))) \quad (24)$$

$$- \frac{\partial}{\partial \mu_k} (\sum_{j \in \mathcal{B}_k} \hat{y}_j^\top \log(\frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp(-\frac{1}{2}(\mathbf{x}_j - \mu_k)^\top \Sigma^{-1}(\mathbf{x}_j - \mu_k)))) \quad (25)$$

$$= -z_{i,k} \Sigma^{-1}(\mathbf{x}_i - \mu_k) + \frac{\partial}{\partial \mu_k} (\beta \text{KL}(\mathcal{N}(\hat{\mu}_k, \hat{\Sigma}_k) \parallel \mathcal{N}(\mu_k, \Sigma))) \quad (26)$$

$$- \sum_{j \in \mathcal{B}_k} \hat{y}_{j,k} \Sigma^{-1}(\mathbf{x}_j - \mu_k) \quad (27)$$

$$\stackrel{(c)}{=} -z_{i,k} \Sigma^{-1}(\mathbf{x}_i - \mu_k) - \sum_{j \in \mathcal{B}_k} \hat{y}_{j,k} \Sigma^{-1}(\mathbf{x}_j - \mu_k) \quad (28)$$

$$+ \frac{\partial}{\partial \mu_k} \beta \frac{1}{2} (\log \frac{|\Sigma|}{|\hat{\Sigma}_k|} - d + (\hat{\mu}_k - \mu_k)^\top \Sigma^{-1}(\hat{\mu}_k - \mu_k) + \text{Tr}(\Sigma^{-1} \hat{\Sigma}_k)) \quad (29)$$

$$= -z_{i,k} \Sigma^{-1}(\mathbf{x}_i - \mu_k) - \sum_{j \in \mathcal{B}_k} \hat{y}_{j,k} \Sigma^{-1}(\mathbf{x}_j - \mu_k) - \beta \Sigma^{-1}(\hat{\mu}_k - \mu_k). \quad (30)$$

43 We obtain $\stackrel{(c)}{=}$ from Eq. (15). And, setting the partial derivative to zero, we can get:

$$\mu_k = \frac{z_{i,k} \mathbf{x}_i + \sum_{j \in \mathcal{B}_k} \hat{y}_{j,k} \mathbf{x}_j + \beta \hat{\mu}_k}{z_{i,k} + \sum_{j \in \mathcal{B}_k} \hat{y}_{j,k} + \beta} \quad (31)$$

$$\stackrel{(d)}{=} \frac{\sum_{j \in \mathcal{B}_k} \hat{y}_{j,k} \mathbf{x}_j + \beta \hat{\mu}_k}{\sum_{j \in \mathcal{B}_k} \hat{y}_{j,k} + \beta} \quad (32)$$

$$= \mu_k^*. \quad (33)$$

44 To enable efficient one-pass, closed-form online inference without iterative updates, we estimate
 45 the class mean by excluding the current instance \mathbf{x}_i in $\stackrel{(d)}{=}$, helping prevent early overfitting and
 46 eliminating sub-iteration. The final mean vector is then expressed as a weighted combination with a
 47 scalar $\alpha \in [0, 1]$:

$$\mu_k^* \leftarrow \alpha \mu_k' + (1 - \alpha) \hat{\mu}_k, \quad \text{where } \mu_k' = \frac{\sum_{j \in \mathcal{B}_k} \hat{y}_{j,k} \mathbf{x}_j}{\sum_{j \in \mathcal{B}_k} \hat{y}_{j,k}}, \alpha = \frac{\sum_{j \in \mathcal{B}_k} \hat{y}_{j,k}}{\sum_{j \in \mathcal{B}_k} \hat{y}_{j,k} + \beta}. \quad (34)$$

48 A.2.2 Proof of Our Closed-form Solution

49 We propose a closed-form label estimator (Eq. (8) in our main paper) for streaming scenarios by
 50 minimizing a regularized objective in Eq. (9). To derive a one-pass solution suitable for online

inference, we compute the partial derivative of the objective with respect to the soft label assignment z_i . Since the Laplacian regularization term in $\mathcal{R}(z_i; \hat{y}_i)$ is concave, we incorporate the simplex constraint $z_i \in \Delta^K$ to ensure that the solution lies within the probability simplex.

$$\frac{\partial \mathcal{L}_{\text{online}}}{\partial z_{i,k}} = \frac{\partial}{\partial z_{i,k}} (-z_i^\top \log \mathbb{P}_i) + \frac{\partial}{\partial z_{i,k}} \text{KL}(z_i \| \hat{y}_i) \quad (35)$$

$$- \frac{\partial}{\partial z_{i,k}} \sum_{j \in \mathcal{B}} w_{ij} z_i^\top \hat{y}_j + \frac{\partial}{\partial z_{i,k}} \lambda_i (\mathbb{I}_K^\top z_i - 1) \quad (36)$$

$$= -\log \mathbb{P}_{i,k} + \log z_{i,k} - \log \hat{y}_{i,k} + 1 - \sum_{j \in \mathcal{B}_k} w_{ij} \hat{y}_{j,k} + \lambda_i \quad (37)$$

$$= -\tilde{y}_{i,k} + \log z_{i,k} - \log \hat{y}_{i,k} + 1 - \sum_{j \in \mathcal{B}_k} w_{ij} \hat{y}_{j,k} + \lambda_i. \quad (38)$$

The first term in Eq. (37) is replaced by $-\tilde{y}_{i,k}$ because we can obtain $\tilde{y}_{i,k} \propto \log \mathbb{P}_{i,k}$ from Section A.1. And, setting the partial derivative to zero, we can get:

$$z_{i,k} = \hat{y}_{i,k} \cdot \exp \left(\tilde{y}_{i,k} + \sum_{j \in \mathcal{B}_k} w_{ij} \hat{y}_{j,k} - (1 + \lambda_i) \right), \quad (39)$$

$$\text{s.t.} \quad \sum_{k=1}^K z_{i,k} = 1. \quad (40)$$

We can get $\exp(-(1 + \lambda_i)) = 1 / (\sum_{k=1}^K \hat{y}_{i,k} \cdot \exp(\tilde{y}_{i,k} + \sum_{j \in \mathcal{B}_k} w_{ij} \hat{y}_{j,k}))$. Bring it into Eq. (39), we can obtain the final optimal solution:

$$z_{i,k}^* = \frac{\hat{y}_{i,k} \cdot \exp \left(\tilde{y}_{i,k} + \sum_{j \in \mathcal{B}_k} w_{ij} \hat{y}_{j,k} \right)}{\sum_{k=1}^K \hat{y}_{i,k} \cdot \exp \left(\tilde{y}_{i,k} + \sum_{j \in \mathcal{B}_k} w_{ij} \hat{y}_{j,k} \right)}. \quad (41)$$

A.3 Transductive Test-time Distribution Estimation

In the transductive setting, the entire test set $\mathcal{D}_u = \{\mathbf{x}_i\}_{i=1}^N$ is available during inference, allowing us to jointly optimize label distributions over all test instances rather than updating them sequentially. We extend the online regularized objective in Eq. (9) to a transductive objective as follows:

$$\mathcal{L}_{\text{trans}}(z, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\sum_{i=1}^N z_i^\top \log \mathbb{P}_i + \sum_{i=1}^N \mathcal{R}(z_i; \hat{y}_i) + \sum_{i=1}^N \mathcal{R}(z_i; \mathcal{B}) \quad (42)$$

$$= -\sum_{i=1}^N z_i^\top \log \mathbb{P}_i + \sum_{i=1}^N \text{KL}(z_i \| \hat{y}_i) + \beta \sum_{k=1}^K \text{KL} \left(\mathcal{N}(\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k) \| \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}) \right) \quad (43)$$

$$- \sum_{j \in \mathcal{B}} \hat{y}_j^\top \log \mathbb{P}_j - \sum_{i=1}^N \sum_{j \in \mathcal{B}} w_{ij} z_i^\top \hat{y}_j. \quad (44)$$

Similar to the online case, we obtain the closed-form label estimator by taking the derivative of $\mathcal{L}_{\text{trans}}$ w.r.t. $z_{i,k}$:

$$\frac{\partial \mathcal{L}_{\text{trans}}}{\partial z_{i,k}} = \frac{\partial}{\partial z_{i,k}} \left(-\sum_{i=1}^N z_i^\top \log \mathbb{P}_i + \sum_{i=1}^N \mathcal{R}(z_i; \hat{y}_i) + \sum_{i=1}^N \mathcal{R}(z_i; \mathcal{B}) \right) \quad (45)$$

$$= \frac{\partial}{\partial z_{i,k}} (-z_i^\top \log \mathbb{P}_i + \mathcal{R}(z_i; \hat{y}_i) + \mathcal{R}(z_i; \mathcal{B})) = \frac{\partial \mathcal{L}_{\text{online}}}{\partial z_{i,k}}. \quad (46)$$

With the results of Section A.2.2, we have:

$$z_{i,k}^* = \frac{\hat{y}_{i,k} \cdot \exp \left(\tilde{y}_{i,k} + \sum_{j \in \mathcal{B}_k} w_{ij} \hat{y}_{j,k} \right)}{\sum_{k=1}^K \hat{y}_{i,k} \cdot \exp \left(\tilde{y}_{i,k} + \sum_{j \in \mathcal{B}_k} w_{ij} \hat{y}_{j,k} \right)}. \quad (47)$$

Then, μ_k is estimated by taking the derivative of $\mathcal{L}_{\text{trans}}$ w.r.t. μ_k :

$$\frac{\partial \mathcal{L}_{\text{trans}}}{\partial \mu_k} = -\frac{\partial}{\partial \mu_k} (\sum_{i=1}^N z_i^\top \log \mathbb{P}_i) + \frac{\partial}{\partial \mu_k} (\beta \sum_{k=1}^K \text{KL}(\mathcal{N}(\hat{\mu}_k, \hat{\Sigma}_k) \parallel \mathcal{N}(\mu_k, \Sigma))) \quad (48)$$

$$-\frac{\partial}{\partial \mu_k} (\sum_{j \in \mathcal{B}} \hat{y}_j^\top \log \mathbb{P}_j) \quad (49)$$

$$= -\sum_{i=1}^N \frac{\partial}{\partial \mu_k} (z_i^\top \log \mathbb{P}_i) + \frac{\partial}{\partial \mu_k} (\beta \sum_{k=1}^K \text{KL}(\mathcal{N}(\hat{\mu}_k, \hat{\Sigma}_k) \parallel \mathcal{N}(\mu_k, \Sigma))) \quad (50)$$

$$-\frac{\partial}{\partial \mu_k} (\sum_{j \in \mathcal{B}} \hat{y}_j^\top \log \mathbb{P}_j) \quad (51)$$

$$= -\sum_{i=1}^N z_{i,k} \Sigma^{-1} (\mathbf{x}_i - \mu_k) - \beta \Sigma^{-1} (\hat{\mu}_k - \mu_k) - \sum_{j \in \mathcal{B}_k} \hat{y}_{j,k} \Sigma^{-1} (\mathbf{x}_j - \mu_k). \quad (52)$$

Setting the Eq. (52) to zero, we have:

$$\mu_k = \frac{\sum_{i=1}^N z_{i,k} \mathbf{x}_i + \sum_{j \in \mathcal{B}_k} \hat{y}_{j,k} \mathbf{x}_j + \beta \hat{\mu}_k}{\sum_{i=1}^N z_{i,k} + \sum_{j \in \mathcal{B}_k} \hat{y}_{j,k} + \beta} \quad (53)$$

$$\stackrel{(e)}{=} \frac{\sum_{i=1}^N \hat{y}_i \mathbf{x}_i + \sum_{j \in \mathcal{B}_k} \hat{y}_{j,k} \mathbf{x}_j + \beta \hat{\mu}_k}{\sum_{i=1}^N \hat{y}_i + \sum_{j \in \mathcal{B}_k} \hat{y}_{j,k} + \beta} = \mu_k^*. \quad (54)$$

In practice, to further improve efficiency and avoid the iterative updates required by MM-like algorithms, we obtain μ_k^* by substituting z_i with the CLIP zero-shot predictions \hat{y}_i , yielding a one-pass estimate for class means:

$$\mu_k^* \leftarrow \alpha \mu_k' + (1 - \alpha) \hat{\mu}_k, \mu_k' = \frac{\sum_{i=1}^N \hat{y}_{i,k} \mathbf{x}_i + \sum_{j \in \mathcal{B}_k} \hat{y}_{j,k} \mathbf{x}_j}{\sum_{i=1}^N \hat{y}_{i,k} + \sum_{j \in \mathcal{B}_k} \hat{y}_{j,k}}, \alpha = \frac{\sum_{i=1}^N \hat{y}_{i,k} + \sum_{j \in \mathcal{B}_k} \hat{y}_{j,k}}{\sum_{i=1}^N \hat{y}_{i,k} + \sum_{j \in \mathcal{B}_k} \hat{y}_{j,k} + \beta}. \quad (55)$$

A.4 Limitations and Discussion

Limitation. Our method assumes that class-conditional features follow Gaussian distributions with a shared covariance matrix. While this assumption simplifies the underlying data distribution and may not fully capture complex, multi-modal, or highly skewed patterns in real-world scenarios, it enables a key practical benefit. Specifically, the Gaussian assumption allows us to derive closed-form, backpropagation-free updates for both online and transductive test-time adaptation. This property is important for deploying vision-language models in environments that require low latency or have limited computational resources. Examples include mobile devices, robotic systems, and real-time inference settings, where gradient-based optimization is often infeasible.

Discussion. To verify the core assumptions of our method that class-conditional features follow Gaussian distributions with a shared covariance matrix, we conduct two empirical analyses.

First, to examine the Gaussianity of class-conditional features, we adopt a projection-based testing strategy inspired by prior statistical literatures [14, 1, 2]. Classical multivariate normality tests [6, 18, 17] are known to break down when the feature dimension is large relative to the sample size (*e.g.*, $512/50 \gg 0$ in our case using CLIP-ViT/B-16 on ImageNet). To address this, we project high-dimensional data into low-dimensional subspaces by randomly selecting a small number of feature dimensions per class, and

Table 1: Projection-based normality test results across class-conditional features.

	Low-dim	Freq of $p > 0.05$ (%) \uparrow	p-value Avg. \uparrow
Henze-Zirkler	2	100	0.39
	4	99.90	0.32
	6	99.00	0.27
	8	96.30	0.22
	10	92.90	0.19
Shapiro-Wilk	2	100	0.31
	4	100	0.21
	6	99.50	0.16
	8	96.30	0.13
	10	92.20	0.11

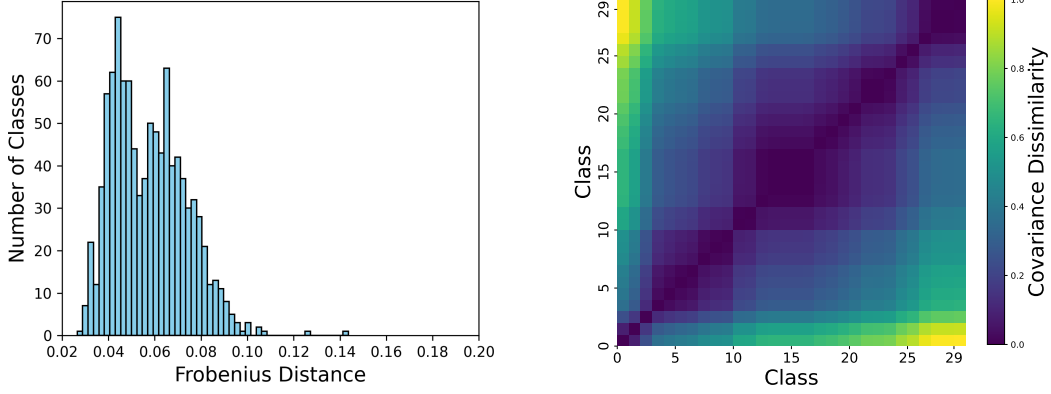


Figure 1: Comparison of covariance properties on ImageNet: (Left) shows Frobenius distance between class-wise Σ_k and shared Σ , and (Right) shows class-wise covariance dissimilarity.

repeat this procedure 100 times. For each projected subset, we apply the Henze–Zirkler test [6] and Shapiro–Wilk test [17] for normality and report the average p-values across repetitions. As shown in Table 1, the average p-values frequently exceed 0.05, suggesting that class-conditional CLIP features are approximately Gaussian in projected subspaces. Furthermore, even when the data deviates from strict Gaussianity, our method remains effective as evidenced in Table 4, where it consistently achieves state-of-the-art performance. Existing works [11, 19, 21] also support the robustness of Gaussian discriminant analysis (GDA) against slight normality violations.

Second, we assess the validity of the shared covariance assumption. For each class, we compute its empirical covariance matrix and compare it to the pooled covariance using the Frobenius norm. As shown in Figure 1 (left), over 99.3% of the classes exhibit a Frobenius distance smaller than 0.1 from the pooled covariance, indicating strong alignment. Additionally, we visualize covariance matrices from several randomly selected classes in Figure 1 (right), and further quantify dissimilarity by comparing the trace values of class-wise covariance matrices. These heatmaps reveal consistent structural patterns across classes, providing qualitative evidence for the shared covariance structure. In summary, these findings collectively provide strong empirical support for the Gaussianity and shared covariance assumptions underlying our method.

To further compare with class-wise separate covariance matrices, we conducted additional experiments on ImageNet by replacing our shared covariance with class-specific covariance matrices, keeping all other settings identical. The results, summarized in Table 2, show that this leads to a significant drop in accuracy, alongside a sharp increase in computation time and memory usage. We attribute this performance degradation to data sparsity. Estimating a full covariance matrix for each class becomes unreliable with few high-confidence samples, leading to poor generalization and overfitting. In contrast, the shared covariance pools information across all classes, resulting in a much more stable and robust estimate, which is critical in online or data-scarce settings. Ultimately, this ablation confirms that our shared covariance assumption is not a limitation, but a crucial design choice. It strikes an effective trade-off between accuracy, robustness, and computational efficiency, making our method practical for real-world, resource-constrained scenarios.

Table 2: Evaluation with class-wise separate covariance matrices on ImageNet.

Method	Time ↓	Acc (%) ↑	Gain (%) ↑	Mem.(GB) ↓
CLIP	8m	66.74	-	0.79
ADAPT (Online)	1h 11m	70.91	+4.17	0.93
w/ Separate Cov.	1h 44m	53.80	-12.94	2.89
ADAPT (Trans.)	0.73m	71.56	+4.82	3.37
w/ Separate Cov.	1.40m	66.16	-0.58	4.62

Remark. To balance efficiency and stability in streaming scenarios with evolving data distributions, we exclude the current test sample \mathbf{x}_i from online class mean updates. This decision is based on three key observations: (i) high-confidence samples are stored in the knowledge bank and will be incorporated later; (ii) low-confidence predictions contribute little and add noise; and (iii) moderately confident predictions are prone to errors, distorting class statistics. By omitting \mathbf{x}_i , we prevent early-stage prediction errors from propagating into the model. As shown in Table 8 (see main paper),

135 this approach offers significant computational savings with minimal accuracy loss, especially in the
 136 early stages of high uncertainty. In the transductive setting, we estimate class means μ_k using the
 137 entire test set and accumulated high-confidence samples. Instead of relying on latent assignments
 138 z_i optimized through iterative procedures (as in MM-based methods), we replace them with CLIP’s
 139 zero-shot predictions \hat{y}_i . This design is motivated by three factors: (i) it avoids costly sub-iterations,
 140 enabling a one-pass, closed-form estimation of μ_k ; (ii) with all test samples available in advance,
 141 computing \hat{y}_i for the entire set is efficient and supports globally consistent estimation; (iii) although
 142 noisy, CLIP’s soft predictions provide a reasonable proxy for class membership when aggregated.
 143 As shown in our ablations (Table 8 in the main paper), this heuristic achieves comparable or better
 144 performance than MM-based optimization, while being more stable and computationally efficient.

145 **Comparison with Other Distribution Estimation Methods.** We compare existing methods for
 146 estimating test-time distributions and demonstrate their limitations or inapplicability under the
 147 realistic online setting.

- 148 • GDA-CLIP [22]: This method estimates the class-wise distribution by computing the empirical
 149 mean and inverse covariance from an external training set $\mathcal{D}_s = \{\mathbf{x}_j\}_{j=1}^S$. The class mean is
 150 calculated as:

$$\mu_k = \frac{\sum_{j \in \mathcal{D}_s} \mathbb{I}_{(y_j=k)} \mathbf{x}_j}{\sum_{j \in \mathcal{D}_s} \mathbb{I}_{(y_j=k)}}. \quad (56)$$

151 Although the covariance is estimated using an empirical Bayes ridge-type estimator, this ap-
 152 proach performs well only when the test distribution closely resembles the source data in its
 153 statistical properties. Crucially, it inherently assumes access to a labeled and stationary source
 154 distribution, which is unavailable in the TTA setting. As a result, it fails to adapt under distri-
 155 bution shifts, and the estimated statistics can quickly become outdated or misaligned with the
 156 continuously evolving test data stream.

- 157 • Frolic [30]: This method directly adopts CLIP’s class prototypes as class-wise means, *i.e.*,
 158 $\mu_k = \mathbf{t}_k$, and estimates the shared covariance Σ from the marginal distribution via the expectation
 159 and second-order moment, as follows:

$$\Sigma = \frac{1}{N} \sum_{i \in \mathcal{D}_u} \mathbf{x}_i \mathbf{x}_i^\top - \frac{1}{K} \sum_{k=1}^K \mu_k \mu_k^\top. \quad (57)$$

160 However, this method requires complete access to the entire target dataset $\mathcal{D}_u = \{\mathbf{x}_i\}_{i=1}^N$, making
 161 it unsuitable for online or streaming scenarios where test samples arrive sequentially and future
 162 data is inaccessible.

- 163 • TransCLIP [26]: This method estimates the parameters μ_k and Σ via a Majorize-Minimize
 164 (MM) optimization procedure, which iteratively refines the predicted soft-assignments z and the
 165 distribution parameters:

$$\mu_k = \frac{\sum_{i \in \mathcal{D}_u} z_{i,k} \mathbf{x}_i}{\sum_{i \in \mathcal{D}_u} z_{i,k}}, \quad (58)$$

$$\text{diag}(\Sigma) = \frac{\sum_{i \in \mathcal{D}_u} z_{i,k} (\mathbf{x}_i - \mu_k)^2}{N}. \quad (59)$$

166 It simultaneously relies on full access to the entire target dataset \mathcal{D}_u and involves computationally
 167 expensive inner-loop MM iterations to achieve convergence. These two limitations significantly
 168 hinder its practicality in real-time or resource-constrained online scenarios, where access to
 169 future samples is restricted and fast adaptation is essential.

- 170 • ADAPT (Ours): In contrast to the above methods, ADAPT progressively estimates class-wise
 171 means and a shared covariance matrix without requiring supervision or access to source data
 172 (see Section A.2 and Section A.3). This leads to a closed-form, training-free solution that is
 173 naturally compatible with both online and transductive test-time adaptation settings. Moreover,
 174 ADAPT incurs minimal computational overhead and seamlessly adapts to distributional shifts in
 175 streaming scenarios.

Table 3: Top-1 accuracy (%) comparison on natural distribution shift task with CLIP-RN50 backbone under both online and transductive protocols.

	Method	BP-free	ImageNet	ImageNet-A	ImageNet-V	ImageNet-R	ImageNet-S	OOD Avg.	Avg.
Online	CLIP [16]	-	58.16	21.83	51.41	56.15	33.37	40.69	44.18
	TPT [13]	✗	60.74	26.67	54.70	59.11	35.09	43.89	47.26
	DiffTPT [4]	✗	60.80	31.06	55.80	58.80	37.10	45.69	48.71
	C-TPT [24]	✗	60.20	23.40	54.70	58.00	35.10	42.80	46.28
	DMN [28]	✗	63.87	28.57	56.12	61.44	39.84	46.49	49.97
	TPS [20]	✗	62.27	29.80	60.04	55.49	35.74	45.27	48.67
	DPE [27]	✗	63.41	30.15	56.72	63.72	40.03	47.66	50.81
	DynaPrompt [23]	✗	61.56	27.84	55.12	60.63	35.64	44.81	48.16
	BCA [29]	✓	61.81	30.35	56.58	62.89	38.04	46.97	49.93
	TDA [8]	✓	61.35	30.29	55.54	62.58	38.12	46.63	49.58
	Dota [5]	✓	61.82	30.81	55.27	62.81	37.52	46.60	49.65
	ADAPT	✓	62.16	33.08	55.97	62.69	40.21	47.99	50.82
Trans.	TransCLIP [26]	✓	58.00	21.93	51.54	35.15	52.79	40.35	43.88
	ADAPT	✓	62.94	33.72	56.57	63.11	41.19	48.65	51.51

Table 4: Top-1 accuracy (%) comparison on corruption robustness task with CLIP-RN50 backbone under both online and transductive protocols.

	Method	Blur				Weather				Digital				Noise			Avg.
		Defo.	Glas.	Moti.	Zoom	Snow	Fros.	Fog	Brig.	Cont.	Elas.	Pix.	JPEG	Gauss.	Shot	Impu.	
Online	CLIP [16]	9.54	3.40	7.46	12.62	12.29	15.72	22.08	41.69	6.24	4.67	11.01	14.24	2.43	3.07	2.52	11.27
	TPT [13]	8.02	2.74	5.34	10.97	10.59	12.92	16.17	35.67	4.45	3.73	11.56	16.68	1.43	1.94	1.42	9.58
	DiffTPT [4]	10.50	3.90	8.62	12.74	11.31	15.03	19.64	37.73	4.98	4.74	13.58	16.56	2.69	3.59	2.08	11.18
	TDA [8]	9.84	4.4	7.38	13.74	13.74	17.16	23.76	44.16	7.00	5.79	11.24	15.26	2.54	3.26	2.72	12.13
	ADAPT	10.54	4.44	8.57	14.34	13.85	17.84	24.56	45.67	7.76	5.85	11.96	15.86	2.91	3.77	2.92	12.72
Trans.	ZLaP [7]	10.3	3.54	7.99	13.47	13.66	17.15	23.2	44.67	6.55	5.15	11.61	14.23	1.17	2.33	1.65	11.82
	TransCLIP [26]	9.38	4.17	7.14	12.42	11.87	15.04	23.93	44.33	6.45	6.11	11.03	14.85	2.87	3.14	3.13	11.72
	ADAPT	12.26	6.11	10.92	17.02	16.67	20.62	27.77	47.31	9.20	8.56	14.61	18.15	4.05	4.80	4.03	14.81

B More Experimental Results

B.1 Experiments with CLIP-RN50 Backbone

Task 1: Natural Distribution Shift. Table 3 reports the performance on natural distribution shift benchmarks using the CLIP-RN50 backbone under both online and transductive settings. In the online setting, ADAPT outperforms all prior backpropagation-free and optimization-based methods, achieving the highest average accuracy of 50.82%. It even surpasses strong transductive methods such as TransCLIP, despite not accessing the full target set. In the transductive setting, where all test samples are available, ADAPT further improves to 51.51%, exceeding TransCLIP by 7.46%. It also ranks first on OOD benchmarks, demonstrating consistent robustness under domain shifts.

Task 2: Corruption Robustness. We further evaluate the performance on corruption robustness benchmarks, with results presented in Table 4. Using the CLIP-RN50 backbone, ADAPT achieves the highest average accuracy of 12.72% in the online setting, outperforming all prior backpropagation-free and optimization-based methods. In the transductive setting, ADAPT further improves to 14.81% and surpasses all compared methods. It consistently ranks first across all 15 corruption types, demonstrating strong resilience to diverse perturbations and confirming its robustness under severe domain shifts.

Task 3: Fine-Grained Categorization. Finally, we evaluate the ADAPT’s adaptation on fine-grained categorization benchmarks, as shown in Table 5. In the Online setting, ADAPT achieves 62.82% average accuracy, which is competitive with the strongest methods like DMN, while maintaining the advantage of being backpropagation-free. In the transductive setting, ADAPT achieves the highest accuracy of 64.64%, outperforming prior state-of-the-art methods such as TransCLIP and StatA. It ranks at the top in most datasets, demonstrating strong generalization to subtle inter-class differences. These results highlight the effectiveness and robustness of our framework under fine-grained and distribution-shifted conditions.

Table 5: Top-1 accuracy (%) comparison on fine-grained categorization task with CLIP-RN50 backbone under both online and transductive protocols.

	Method	BP-free	Aircraft	Caltech	Cars	DTD	EuroSAT	Flower	Food101	Pets	Sun397	UCF101	Avg.
Online	CLIP [16]	-	15.66	85.88	55.70	40.37	23.69	61.75	73.97	83.57	58.80	58.84	55.82
	TPT [13]	✗	17.58	87.02	58.46	40.84	28.33	62.69	74.88	84.49	61.46	60.82	57.66
	DiffTPT [4]	✗	17.60	86.89	60.71	40.72	41.04	63.53	79.21	83.40	62.72	62.67	59.85
	C-TPT [24]	✗	17.00	86.90	56.50	42.20	27.80	65.20	74.70	84.10	61.00	59.70	57.51
	DMN [28]	✗	22.77	90.14	60.02	50.41	48.72	67.93	76.70	86.78	64.39	65.34	63.32
	TPS [12]	✗	18.30	89.80	59.40	48.40	24.30	68.20	76.20	84.40	62.70	64.30	59.60
	DPE [27]	✗	19.80	90.83	59.26	50.18	41.67	67.60	77.83	85.97	64.23	61.98	61.94
	BCA [29]	✓	19.89	89.70	58.13	48.58	42.12	66.30	77.19	85.58	63.38	63.51	61.44
	TDA [8]	✓	17.61	89.70	57.78	43.74	42.11	68.74	77.75	86.18	62.53	64.18	61.03
	Dota [3]	✓	18.06	88.84	58.72	45.80	47.15	68.53	78.61	87.33	63.89	65.08	62.20
	ADAPT	✓	18.00	89.37	58.38	51.89	50.47	70.04	75.57	86.43	64.94	63.12	62.82
Trans.	TransCLIP [26]	✓	16.60	88.60	57.90	47.80	59.60	72.20	78.00	89.30	64.20	68.80	64.30
	StatA [25]	✓	16.00	87.30	58.20	48.50	50.50	67.70	77.90	87.70	64.30	67.50	62.56
	ADAPT	✓	19.53	90.99	61.46	55.73	46.26	74.14	76.97	87.95	66.66	66.75	64.64

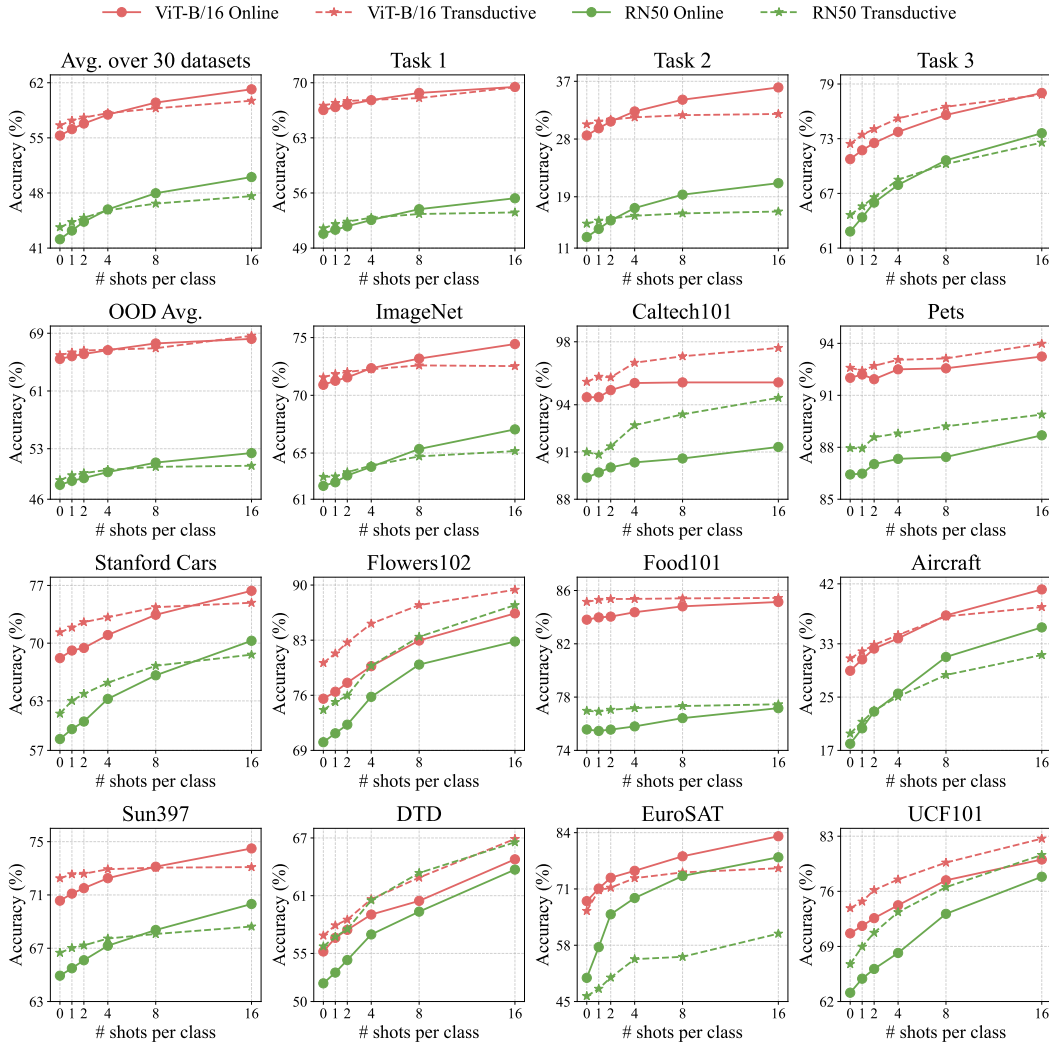


Figure 2: Results of few-shot classification across 30 datasets. We evaluate our method under both online and transductive protocols with 0, 1, 2, 4, 8, and 16-shot settings.

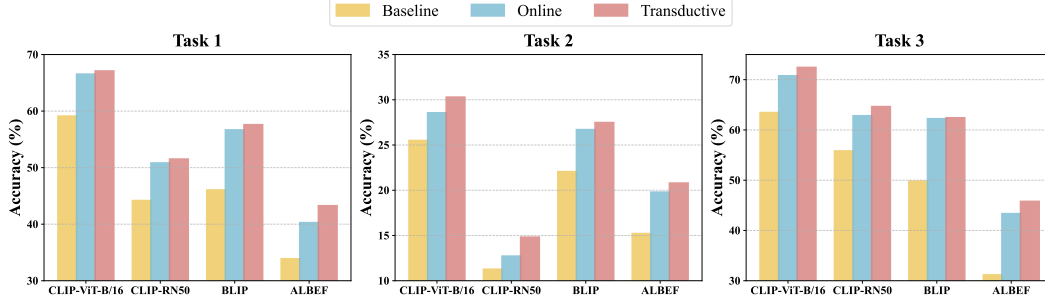


Figure 3: Performance comparison of proposed ADAPT on different VLMs.

B.2 Few-shot Adaptation

Figure 2 presents few-shot classification results of our ADAPT across three tasks covering 30 datasets. We evaluate performance under both online and transductive adaptation protocols, using 0, 1, 2, 4, 8, and 16 shots per class. To assess architectural generality, we conduct experiments with two backbone variants: CLIP-ViT-B/16 and CLIP-RN50.

Overall, results show a consistent performance increase with more labeled samples, confirming the scalability and effectiveness of our method. Results with CLIP-ViT-B/16 consistently outperform RN50 across all shot counts, particularly in low-shot regimes, suggesting stronger feature representations. Additionally, transductive adaptation variants exhibit consistently better performance than their online counterparts. This improvement can be attributed to the transductive setting’s ability to leverage the entire test set as a global context, thereby facilitating more informed label predictions and reducing uncertainty in classification. Notably, Task 3 requires finer-grained discrimination due to subtle intra-class variations, making accurate classification particularly dependent on detailed visual cues. In this context, the few-shot setting offers class-specific supervision that enhances the model’s ability to capture these nuances. As a result, our method exhibits substantial performance gains on Task 3, with consistent improvements observed across 10 representative datasets. These findings underscore our method’s capacity to adapt to complex recognition challenges and its effectiveness across varying adaptation regimes.

B.3 Evaluation with Different VLMs

We conduct a comprehensive evaluation of our ADAPT across a diverse set of vision-language models (VLMs), including CLIP (ViT-B/16 and RN50) [16], BLIP [9], and ALBEF [10]. Figure 3 presents the performance comparison across three representative tasks: Task 1 assessing natural distribution shifts, Task 2 evaluating robustness to synthetic corruptions, and Task 3 focusing on fine-grained categorization challenges.

Our results demonstrate that ADAPT consistently improves performance over baseline methods across all tasks and model architectures. The online adaptation variant already delivers notable gains by leveraging sequential test data, while the transductive variant further enhances results by exploiting global information from the entire test set. Importantly, these improvements hold true not only for stronger VLMs like CLIP-ViT-B/16 but also for comparatively lighter models such as ALBEF, indicating that ADAPT is broadly applicable and effective regardless of the underlying model capacity. This versatility highlights the practical value of our approach for enhancing test-time adaptation across diverse vision-language architectures and real-world scenarios.

B.4 Additional Analysis

Further Analysis on Ablation Study. Table 6 presents a detailed ablation study of three key components in our ADAPT: (i) the knowledge bank \mathcal{B} , (ii) the adaptive class-wise mean μ , and (iii) shared covariance Σ . We compare eight configurations by selectively enabling or disabling these components and report performance across all three tasks. In this setup, disabling μ means using CLIP’s original class prototypes as the fixed class means, while disabling Σ corresponds to using a fixed identity matrix as the shared covariance.

The upper block of the table (Rows 1–4) corresponds to knowledge bank-free settings, where \mathcal{B} is not used. The first row represents the baseline CLIP model without any adaptation. Updating only μ (Row 3) provides modest improvements (*e.g.*, +2.43% on Task 1), likely due to better-aligned class centers. However, updating only Σ (Row 2) leads to substantial performance drops (*e.g.*, down to 9.58% on Task 2), indicating that estimating covariance from noisy test-time predictions alone is highly unstable and unreliable.

The lower block (Rows 5–8) introduces the knowledge bank \mathcal{B} . Even without updating μ or Σ (Row 5), the model significantly outperforms all memory-free variants, validating the effectiveness of the regularization term $\mathcal{R}(z_i; \mathcal{B})$, which encourages alignment with high-confidence stored features. Incorporating adaptation for either μ or Σ further improves results, showing that both distributional statistics offer complementary cues for more accurate estimation. The best performance is achieved by jointly adapting both μ and Σ , which fully leverages the knowledge bank for robust distribution modeling, with Task 2 reaching 28.56%. These results highlight the synergy between adaptive statistics and memory-guided estimation for stable TTA.

Table 6: Ablation study on components.

\mathcal{B}	Update μ	Update Σ	Task 1	Task 2	Task 3
\times	\times	\times	59.11	25.50	63.45
\times	\times	\checkmark	49.64	9.58	60.02
\times	\checkmark	\times	61.54	25.42	67.03
\times	\checkmark	\checkmark	49.65	9.58	60.04
\checkmark	\times	\times	64.89	25.08	67.06
\checkmark	\times	\checkmark	64.05	19.49	67.95
\checkmark	\checkmark	\times	65.27	25.67	67.43
\checkmark	\checkmark	\checkmark	66.53	28.56	70.76

Robustness under Significant Shifts.

To evaluate our method’s robustness under significant distribution shifts, we benchmarked ADAPT against TDA [8], a state-of-the-art memory-based method. The evaluation was conducted using synthetic corruptions across five levels of increasing severity, with detailed results presented in Table 7. The results show that while all methods degrade as the shift intensifies, ADAPT consistently outperforms TDA across all severity levels. This provides strong evidence that our proposed mechanism is more resilient to noisy pseudo-labels, validating its superior robustness for practical applications involving severe, real-world distribution shifts.

Table 7: Robustness evaluation on ImageNet-C across five levels of synthetic corruption.

Severity Level	1	2	3	4	5
CLIP	59.68	52.97	46.79	37.51	25.50
TDA	60.42	54.15	48.35	36.84	28.34
ADAPT (Online)	61.37	54.70	48.61	39.28	28.56
ADAPT (Trans.)	62.04	55.51	49.68	40.76	30.29

Robustness under Low-confidence Scenarios.

To assess the impact of having a limited number of high-confidence samples per class (denoted as N), we analyze our method’s performance in this constrained setting. The results, presented in Table 8, reveal distinct behaviors for our online and transductive approaches. Our online method, while sensitive to extremely low sample counts ($N = 2$), recovers quickly and achieves robust performance with as few as 4–6 samples per class. However, we acknowledge a potential efficiency challenge: if the test stream begins with many low-confidence predictions, it takes longer to collect reliable samples for the memory bank, which may slightly delay adaptation and degrade performance. Notably, our transductive ADAPT demonstrates exceptional robustness, maintaining high and stable performance even with just two high-confidence samples per class ($N = 2$). We attribute this resilience to its ability to leverage the global structure of the entire test batch, which regularizes parameter estimates and ensures strong performance even under severe data scarcity.

Table 8: Evaluation with very few high-confidence samples. Where N means the number of high-confidence samples per class.

	N	2	4	6	8	10
Online	Task 1	59.61	64.19	65.50	66.10	66.33
	Task 2	22.95	26.00	27.26	27.93	28.22
	Task 3	63.87	67.46	69.04	69.89	70.31
Trans.	Task 1	67.02	67.11	67.09	67.10	67.04
	Task 2	30.26	30.29	30.29	30.23	30.23
	Task 3	72.15	72.21	72.43	72.24	72.25

References

- [1] Sara El Bouch, Olivier Michel, and Pierre Comon. A normality test for multivariate dependent samples. *Signal Processing*, 201:108705, 2022.
- [2] Sara ElBouch, Olivier JJ Michel, and Pierre Comon. Joint normality test via two-dimensional projection. In *ICASSP*, 2022.
- [3] Matteo Farina, Gianni Franchi, Giovanni Iacca, Massimiliano Mancini, and Elisa Ricci. Frustratingly easy test-time adaptation of vision-language models. In *NeurIPS*, 2024.
- [4] Chun-Mei Feng, Kai Yu, Yong Liu, Salman Khan, and Wangmeng Zuo. Diverse data augmentation with diffusions for effective test-time prompt tuning. In *ICCV*, 2023.
- [5] Zongbo Han, Jialong Yang, Junfan Li, Qinghua Hu, Qianli Xu, Mike Zheng Shou, and Changqing Zhang. Dota: Distributional test-time adaptation of vision-language models. *arXiv preprint arXiv:2409.19375*, 2024.
- [6] Norbert Henze and Bernd Zirkler. A class of invariant consistent tests for multivariate normality. *Communications in statistics-Theory and Methods*, 19(10):3595–3617, 1990.
- [7] Yannis Kalantidis, Giorgos Tolias, et al. Label propagation for zero-shot classification with vision-language models. In *CVPR*, 2024.
- [8] Adilbek Karmanov, Dayan Guan, Shijian Lu, Abdulmotaleb El Saddik, and Eric Xing. Efficient test-time adaptation of vision-language models. In *CVPR*, 2024.
- [9] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In *ICML*, 2022.
- [10] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align Before Fuse: Vision and Language Representation Learning with Momentum Distillation. In *NeurIPS*, 2021.
- [11] Tao Li, Shenghuo Zhu, and Mitsunori Ogihara. Using discriminant analysis for multi-class classification: an experimental investigation. *Knowledge and information systems*, 10:453–472, 2006.
- [12] Yushu Li, Yongyi Su, Adam Goodge, Kui Jia, and Xun Xu. Efficient and context-aware label propagation for zero-/few-shot training-free adaptation of vision-language model. In *ICLR*, 2025.
- [13] Shu Manli, Nie Weili, Huang De-An, Yu Zhiding, Goldstein Tom, Anandkumar Anima, and Xiao Chaowei. Test-time prompt tuning for zero-shot generalization in vision-language models. In *NeurIPS*, 2022.
- [14] Alicia Nieto-Reyes, Juan Antonio Cuesta-Albertos, and Fabrice Gamboa. A random-projection based test of gaussianity for stationary processes. *Computational Statistics & Data Analysis*, 75:124–141, 2014.
- [15] Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.
- [16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.
- [17] J Patrick Royston. An extension of shapiro and wilk’s w test for normality to large samples. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 31(2):115–124, 1982.
- [18] S Shaphiro and MBBJ Wilk. An analysis of variance test for normality. *Biometrika*, 52(3):591–611, 1965.
- [19] Housseem Sifaou, Abba Kammoun, and Mohamed-Slim Alouini. High-dimensional linear discriminant analysis classifier for spiked covariance model. *Journal of Machine Learning Research*, 21(112):1–24, 2020.

- 341 [20] Elaine Sui, Xiaohan Wang, and Serena Yeung-Levy. Just shift it: Test-time prototype shifting
342 for zero-shot generalization with vision-language models. In *WACV*. IEEE, 2025.
- 343 [21] Raquel Urtasun and Trevor Darrell. Discriminative gaussian process latent variable model for
344 classification. In *ICML*, 2007.
- 345 [22] Zhengbo Wang, Jian Liang, Lijun Sheng, Ran He, Zilei Wang, and Tieniu Tan. A hard-to-beat
346 baseline for training-free clip-based adaptation. In *ICLR*, 2024.
- 347 [23] Zehao Xiao, Shilin Yan, Jack Hong, Jiayin Cai, Xiaolong Jiang, Yao Hu, Jiayi Shen, Qi Wang,
348 and Cees GM Snoek. Dynaprompt: Dynamic test-time prompt tuning. In *ICLR*, 2025.
- 349 [24] Hee Suk Yoon, Eunseop Yoon, Joshua Tian Jin Tee, Mark Hasegawa-Johnson, Yingzhen Li,
350 and Chang D Yoo. C-tpt: Calibrated test-time prompt tuning for vision-language models via
351 text feature dispersion. In *ICLR*, 2024.
- 352 [25] Maxime Zanella, Clément Fuchs, Christophe De Vleeschouwer, and Ismail Ben Ayed. Realistic
353 test-time adaptation of vision-language models. In *CVPR*, 2025.
- 354 [26] Maxime Zanella, Benoît Gérin, and Ismail Ayed. Boosting vision-language models with
355 transduction. In *NeurIPS*, 2024.
- 356 [27] Ce Zhang, Simon Stepputtis, Katia Sycara, and Yaqi Xie. Dual prototype evolving for test-time
357 generalization of vision-language models. In *NeurIPS*, 2024.
- 358 [28] Yabin Zhang, Wenjie Zhu, Hui Tang, Zhiyuan Ma, Kaiyang Zhou, and Lei Zhang. Dual memory
359 networks: A versatile adaptation approach for vision-language models. In *CVPR*, 2024.
- 360 [29] Lihua Zhou, Mao Ye, Shuaifeng Li, Nianxin Li, Xiatian Zhu, Lei Deng, Hongbin Liu, and Zhen
361 Lei. Bayesian test-time adaptation for vision-language models. In *CVPR*, 2025.
- 362 [30] Xingyu Zhu, Beier Zhu, Yi Tan, Shuo Wang, Yanbin Hao, and Hanwang Zhang. Enhancing
363 zero-shot vision models by label-free prompt distribution learning and bias correcting. In
364 *NeurIPS*, 2024.