Space Rotation with Basis Transformation for Training-free Test-Time Adaptation

Anonymous Author(s)
Affiliation
Address
email

Abstract

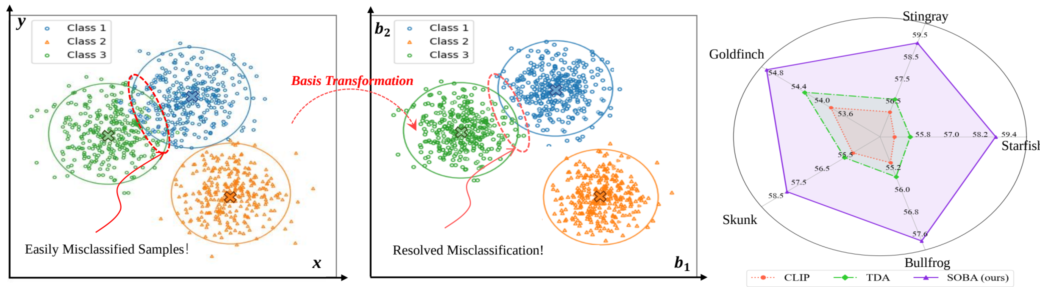
With the widespread application of Vision-Language Models (VLMs) in downstream tasks, test-time adaptation methods based on VLMs, particularly the training-free paradigm, have been gaining increasing attention due to their advantages in handling distribution shifts during testing. Although existing training-free methods have made some progress, their performance remains suboptimal due to the limitations of the original feature space. To address this issue, we propose a training-free feature space rotation with basis transformation for test-time adaptation. Inspired by classical machine learning theories, we construct the orthogonal basis by leveraging the inherent differences among classes and reconstruct the original feature space through basis transformation, leading to clearer decision boundaries. Our approach significantly enhances class discriminability and provides more effective guidance for the model during testing. Experimental results across multiple benchmarks demonstrate that our method outperforms state-of-the-art methods in terms of both performance and efficiency.

1 Introduction

Visual-language models (VLMs), such as CLIP [1] and ALIGN [2], have garnered significant attention due to their strong generalization capabilities in downstream tasks. Currently, various efficient tuning methods, such as prompt tuning [3, 4, 5] and adapter tuning [6, 7], have been proposed to leverage training data for enhancing the performance of VLMs on downstream tasks. While these methods show strong performance, their reliance on training data distribution hinders generalization to new domains. Therefore, test-time adaptation (TTA) [8, 9], which leverages test samples to adjust to downstream data distribution rapidly, holds significant promise for practical applications.

The present mainstream TTA methods for VLMs can be divided into two categories: (i) Prompt-tuning TTA paradigm. TPT [8], DiffTPT [10], and HisTPT [11] tune prompts through different data augmentation and confidence selection strategies, ensuring consistent predictions across different augmented views of each test data. (ii) Training-free TTA paradigm. TDA [9] proposes a training-free dynamic adapter and maintains a high-quality test set to guide the test-time adaptation for VLM. Among them, the prompt-tuning TTA methods [8, 10] demand substantial computational resources and time, contradicting the need for rapid adaptation in real-world scenarios. Therefore, this paper focuses on the **training-free** TTA paradigm.

Despite its decent performance, the training-free TTA method has a significant drawback, which stems from the characteristics of the training-free paradigm. Due to the inability to perform training, adjusting the feature space becomes very difficult, and thus, the effectiveness of the "guidance" entirely depends on the original CLIP feature space. As shown in Fig. 1 (a), the test samples inside the red circle are hard for CLIP to predict accurately due to the overlap of decision boundaries. Currently, methods such as TDA [9], which assist prediction by comparing test samples with representative



(a) Frozen CLIP feature space. $\boldsymbol{x}$ and $\boldsymbol{y}$ are the basis vectors of the original space.

(b) Transformed feature space. $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are the basis vectors of the new space.

(c) Performance comparison on Difficult Classes.

Figure 1: (a) Original CLIP Space. In the original feature space, CLIP may misclassify test samples (red circles) due to overlapping decision boundaries for certain classes. Therefore, for training-free TTA methods, the inability to adjust the feature space limits their applicability in downstream tasks. (b) Feature space after basis transformation. We apply a basis transformation using new vectors (e.g., $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ in Fig.(b)) to the feature space, making it linearly separable. In this transformed space, we establish clearer decision boundaries, addressing the limitation of training-free TTA methods that cannot adjust the feature space. (c) Performance comparison on the difficult classes. Our method demonstrates a more significant improvement over the current SOTA methods in challenging classes.

samples in the original feature space, evidently fail to address this inherent limitation. As shown in Fig. 1 (c), the performance gain of TDA on difficult classes with overlapping decision boundaries is very limited. This raises the following question: *Can we enhance the separability of the feature space without training?*

Classical machine learning provides a novel perspective for this limitation, particularly through the core concepts of support vector machines (SVM) [12] and principal component analysis (PCA) [13]. Specifically, SVM demonstrates that by reconstructing the feature space, low-dimensional nonlinear problems can be transformed into high-dimensional linearly solvable ones; meanwhile, PCA shows that performing a linear orthogonal transformation on the feature matrix can extract the most discriminative principal components. Consequently, mathematical transformations that reconstruct the feature space help reveal the intrinsic discriminability structure of the data, thereby overcoming the limitations of the original feature representation.

Inspired by these theories, we propose a novel training-free test-time adaptation method called **S**pace r**O**tation with **B**asis tr**A**nsformation (SOBA). This approach leverages basis transformation techniques [14] to convert the original linearly non-separable space into a new linearly separable space, thereby optimizing the decision boundary of the original CLIP model and effectively overcoming the limitations inherent in training-free TTA paradigms. Specifically, during testing, pseudo-labels are assigned to each sample based on CLIP's predictions, while a dynamic queue is maintained to store a small set of representative sample features along with their corresponding pseudo-labels. First, we perform singular value decomposition (SVD) on the covariance matrix of the stored representative sample set and construct an orthogonal basis $\mathcal{B}$ through linear transformation to extract the most discriminative information from features across different categories [13]. Second, the original CLIP feature space is reconstructed using the orthogonal basis $\mathcal{B}$, enhancing the linear separability of features in the transformed space [12]. Finally, SOBA computes the mean vectors for each category within this space and employs them as decision boundaries in the new feature space. As shown in Fig. 1 (b), compared to the original CLIP feature space, the transformed space constructed using the basis $\mathcal{B}$ exhibits clearer decision boundaries. This enhances the separability of challenging classes in the new feature space, leading to a significant performance improvement on these classes, as illustrated in Fig. 1 (c).

In this paper, we present three key contributions. First, we analyze the limitations of current training-free TTA methods in adjusting the feature space. Inspired by machine learning theories, we propose a space rotation method based on basis transformation, which reshapes the feature space and effectively solves the issue of inseparability in the original feature space. Second, our method achieves state-of-the-art (SOTA) performance across out-of-distribution and cross-dataset benchmarks, effectively adapting distribution shifts in downstream tasks. Finally, our method also achieves high computational efficiency. Experiments on the ImageNet dataset show that, compared to the training-free SOTA method TDA [9], our approach improves test speed by 13.96% while incurring only 2.15% of the time cost of the fine-tuning-based TPT [8], highlighting its practical applicability.

2 Related Works

Vision-Language Model. In recent years, vision-language models have gained widespread attention for their ability to process both visual and linguistic modalities. Models such as CLIP [1], ALIGN [2], BLIP [15], and FILIP [16] leverage self-supervised training on image-text pairs to establish connections between vision and language, enabling strong semantic understanding. This capability allows vision-language models (e.g., CLIP) to exhibit remarkable generalization across various downstream tasks [17, 18, 19, 20]. Prompt tuning and adapter methods have been introduced to enhance the transferability of vision-language models. However, prompt tuning methods (e.g., CoOp [3], CoCoOp [4], Maple [5]) and adapter-based approaches (e.g., Tip-Adapter [6], CLIP-Adapter [7]) typically require large amounts of training data when adapting to downstream tasks, which limits their applicability in real-world scenarios that demand rapid adaptation. Therefore, this paper focuses on test-time adaptation (TTA) [8], a method that enables model transfer to downstream tasks without relying on training data.

Test-Time Adaptation. TTA enables models to adapt to distribution shifts during inference without training data [21, 22, 23, 24, 25]. TPT [8] learns adaptive prompts via entropy minimization, while DiffTPT [10] enhances diversity using Stable Diffusion [26] and filters augmentations by cosine similarity. Both require backpropagation, limiting efficiency. TDA [9] avoids this by leveraging a cache model [6] to refine predictions via test-sample similarity, enabling training-free adaptation. However, it still operates within CLIP's original feature space. We propose mapping features to a spherical space to better handle distribution shifts. BoostAdapter [27] is excluded due to incompatible settings; results under its setup are in the Appendix (Table D).

Statistical Learning. Statistical learning techniques play an important role in dimensionality reduction and feature extraction. Support Vector Machines (SVM) [12] are primarily used for classification tasks but have been adapted for space mapping through their ability to create hyperplanes that separate data in high-dimensional spaces. The kernel trick enables SVM to operate in transformed feature spaces, effectively mapping non-linearly separable data. PCA [13] is a linear transformation method that maps high-dimensional data to a new lower-dimensional space through a linear transformation, while preserving as much important information from the original data as possible.

3 Method

3.1 A Training-free Baseline

CLIP [1] is a pre-trained vision-language model composed of two parts: a visual encoder and a text encoder, which we represent separately $E_v(\theta_v)$ and $E_t(\theta_t)$. In classification tasks, given a test image x_{test} and *N* classes, CLIP uses $E_t(\theta_t)$ and $E_v(\theta_v)$ to encode handcrafted text descriptions of the *N* classes and x_{test}. After obtaining the corresponding text embeddings $\mathbf{W}_t$ and visual embedding $\boldsymbol{f}_{test}$, CLIP matches the image with the most relevant text description to produce the final prediction as follows:

$$logits_{\mathrm{ori}} = \boldsymbol{f}_{test}\mathbf{W}_t^{\mathrm{T}}. \tag{1}$$

Before starting our method, we first construct a training-free baseline method. We utilize a dynamic queue to store representative samples and use these samples to assist in the prediction of test examples. This prediction is combined with the zero-shot CLIP predictions to produce the final inference. Specifically, we dynamically store $\mathbf{K}$ test examples for each pseudo-classes, along with their corresponding pseudo-labels $\hat{l}$, using minimum entropy as the criterion. Here, the pseudo-labels are obtained by one-hot encoding the predictions $\boldsymbol{f}_{test}{\mathbf{W}_t}^{\mathrm{T}}$ for each sample:

$$\hat{l} = \mathrm{OneHot}(\boldsymbol{f}_{test}{\mathbf{W}_t}^{\mathrm{T}}). \tag{2}$$

When the queue reaches capacity $\mathbf{K}$, we update the queue by replacing the test sample with the highest entropy using the principle of minimizing entropy. Then, during testing, we use an NCM classifier to assist with classification:

$$logits_{\mathrm{NCM}} = \mathrm{sim}(\boldsymbol{f}_{\mathrm{test}}, \mu), \tag{3}$$

where sim is the cosine similarity, and μ is the class mean for each category in the queue.

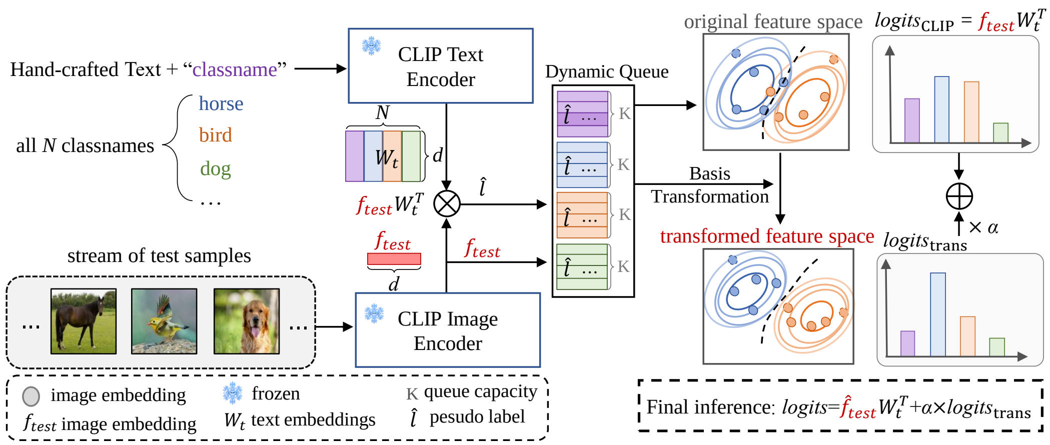


Figure 2: An overview of our method. Our method utilizes a dynamic cache queue to store representative samples and generates predictions for test examples based on these samples. This prediction is combined with zero-shot CLIP predictions to produce the final inference. Specifically, we maintain a dynamic queue of representative samples, selected based on minimum entropy of CLIP's predictions. We construct a basis transformation using these stored samples to facilitate feature space rotation. As testing progresses, we continuously update and utilize these mappings, making the decision boundaries obtained through reconstruction more refined and accurate. Finally, we combine the inferences from CLIP with those from the dynamic queue to obtain the final prediction.

3.2 Theoretical Foundation

During testing, pre-trained models like CLIP often experience reduced generalization due to distribution shifts between downstream tasks and the pre-training dataset. Current approaches focus on improving the selection of augmented views to mitigate this. However, the inference process still faces challenges because the decision boundary remains based on the original CLIP's feature space. For categories with initially poor predictions, the decision boundary in the original feature space limits the effectiveness of augmented view selection, preventing more accurate decisions. This limitation undermines the model's scalability in TTA scenarios.

In this paper, our motivation is to overcome the limitations of the original CLIP feature space for test-time adaptation, aiming to identify a suitable basis. By using the basis to map the original CLIP feature space into a new space, we strive to provide a more effective decision boundary for the inference process. To accomplish this, we propose a training-free feature space rotation method, SOBA, to achieve test-time adaptation of CLIP in downstream tasks.

Before describing our solution, we first present a general explanation of the feature space rotation with basis transformation proposed in this paper. We start by defining a set of feature vectors $W \in \mathbb{R}^{n\times d}$ as a linear combination of standard orthogonal matrices $\mathcal{E} = \{\mathbf{e}_{ij}\}_{i,j}$, where $\mathbf{e}_{ij} \in \mathbb{R}^{n\times d}$ is defined as a matrix with the (i,j)-th element equal to 1 and all other elements equal to 0. Therefore, we can express W as:

$$W = \sum_{i=1}^{n}\sum_{j=1}^{d} w_{ij}\mathbf{e}_{ij}, \tag{4}$$

where, w_{ij} represents the (i,j)-th element of W, which is also the coefficient of $\mathbf{e}_{ij}$. In this paper, we use an arbitrary basis $\mathcal{B} = \{\boldsymbol{b}_{ij} \in \mathbb{R}^{n\times d}\}_{i\in[n],j\in[d]}$ to extend W. Specifically, $\mathcal{B}$ serves as a standard orthogonal basis and must satisfy the following conditions:

$$\begin{aligned} \langle \boldsymbol{b}, \boldsymbol{b}' \rangle &= 0 \text{ if } \boldsymbol{b} \neq \boldsymbol{b}' \text{ for } \boldsymbol{b}, \boldsymbol{b}' \in \mathcal{B}, \\ \|\boldsymbol{b}\| &= \sqrt{\langle \boldsymbol{b}, \boldsymbol{b} \rangle} = 1 \text{ for all } \boldsymbol{b} \in \mathcal{B}, \end{aligned} \tag{5}$$

where, $\|\cdot\|$ and $\langle\cdot\rangle$ represent the norm and inner product, respectively.

Since the vector hilbert space $\mathcal{H} := \mathbb{R}^{n\times d}$ satisfies the inner product operation $\langle \mathrm{C}, \mathrm{D}\rangle = \text{trace}(\mathrm{C}^{\mathrm{T}}\mathrm{D})$ (where $\mathrm{C}, \mathrm{D} \in \mathcal{H}$), we can always express $W \in \mathcal{H}$ as a linear combination of orthogonal matrices in the basis $\mathcal{B}$ under any circumstances. Therefore, Eq. (4) can be expanded into the following form:

$$W = \sum_{\boldsymbol{b}\in\mathcal{B}} \langle W, \boldsymbol{b}\rangle \boldsymbol{b} = \sum_{i=1}^{n}\sum_{j=1}^{d} \langle W, \boldsymbol{b}_{ij}\rangle \boldsymbol{b}_{ij}. \tag{6}$$

We observe that when $\mathcal{B} = \mathcal{E}$, Eq. (6) reduces to Eq. (4). Consequently, when all elements in $\mathcal{B}$ are orthogonal matrices, we can use $\mathcal{B}$ to project W onto a new hypersphere through the mapping $\hat{w} = \{\langle W, \boldsymbol{b}\rangle\}_{\boldsymbol{b}\in\mathcal{B}}$. In Section 3.3, we will describe how to use SOBA to address challenges in the TTA task.

3.3 Space Rotation with Basis Transformation

In this section, we first introduce how to construct an appropriate basis vector matrix using SOBA. Then, we explain how to implement it through parameter estimation.

Basis Construction. To identify an appropriate basis for reconstructing the matrix $W \in \mathbb{R}^{n\times d}$, we begin by defining the basis using a pair of unitary matrices. Let $P \in \mathbb{R}^{n\times n}$ and $Q \in \mathbb{R}^{d\times d}$ be two arbitrary unitary matrices. We observe that the set $\mathcal{B} = \{\boldsymbol{b}_{ij} := p_i q_j^{\mathrm{T}} \in \mathbb{R}^{n\times d}\}_{i\in[n],j\in[d]}$ forms an orthogonal basis, where p_i and q_j represent the i-th column of P and the j-th column of Q, respectively. Consequently, we can express Eq. (6) as follows:

$$W = \sum_{i=1}^{n}\sum_{j=1}^{d} \langle W, \boldsymbol{b}_{ij}\rangle \boldsymbol{b}_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{d} \left\langle W, p_i q_j^{\mathrm{T}}\right\rangle p_i q_j^{\mathrm{T}} = \sum_{i=1}^{n}\sum_{j=1}^{d} \hat{w}_{ij} p_i q_j^{\mathrm{T}}, \tag{7}$$

where $\hat{w} := \left\langle W, p_i q_j^{\mathrm{T}}\right\rangle$. In this case, the basis $\{p_i q_j^{\mathrm{T}}\}_{i,j}$, constructed from a pair of unitary matrices P and Q, maps W into the form of $\hat{w}$. Now, the current challenge is *how to design P and Q to achieve a better basis transformation, thereby obtaining an improved space mapping to address distribution shifts in downstream tasks.*

According to the theory of PCA [13], for a set of feature vectors, we can perform singular value decomposition on their covariance C to extract the main information:

$$C = Q_c \Sigma Q_c^{\mathrm{T}}, \tag{8}$$

where Σ is a diagonal matrix with singular values on its diagonal, and Q_c is the corresponding unitary matrix. As observed in the literature [28], the features obtained from deep neural networks are often low-rank, meaning that most singular values are close to zero. Due to this low-rank property, for any unitary matrix P, setting $Q = Q_c$ allows us to extract important information from W under the basis $\mathcal{B} = \{p_i q_j^{\mathrm{T}}\}$ and map this information to $\hat{w}$. We will introduce how to obtain the covariance matrix C in Eq. (11).

Implementation. Subsequently, we will examine the implementation of our proposed method building upon the foundation of the baseline approach in 3.1. Based on the dynamic queue of the baseline, we utilize SOBA to map the stored features onto a hypersphere, thereby achieving feature reconstruction. The following describes how to implement Eq. (7).

Implementation of W: Similar to the NCM classifier [29], we use the class mean $\mu = \{\mu_k\}_{k=1}^{N}$ from the queue as the classifier weights. Setting $W = \mu$ in Eq. (7) gives us the mapped class mean $\hat{\mu}$. Here, we use the empirical mean to estimate the class mean:

$$\mu_k = \frac{\sum_{i=1}^{M_k} \mathbb{I}_{\hat{l}=k} \boldsymbol{f}_{test,i}}{\sum_{i=1}^{M_k} \mathbb{I}_{\hat{l}=k}}, \tag{9}$$

where, M_k is the total number of class k. $\hat{l}$ is the pseudo-label of samples in the queue.

Implementation of $P = \{p_i\}$ and $Q = \{q_j\}$: In practice, we implement Eq. (7) using a very straightforward approach. Due to the properties of the unitary matrix, we can obtain $PP^{\mathrm{T}} = I_n$ and $QQ^{\mathrm{T}} = I_d$. Then, we express W as following:

$$W = PP^{\mathrm{T}} W QQ^{\mathrm{T}} = P\hat{W}Q. \tag{10}$$

Table 1: **Results on the OOD Benchmark**. Compare the performance of our method with existing methods on OOD benchmark. Our method performs best on both backbones. The best results are in **bold** and the second-best results are <u>underlined</u>. Among the methods we compared, CoOp [3] and CoCoOp [4] are fine-tuned on the training set; TPT [8] and DiffTPT [10] require backpropagation to update the prompts; TDA [9], and our method do not require any backpropagation to update parameters. *OOD average* refers to the average accuracy on the four OOD datasets from ImageNet, while *average* refers to the average accuracy across all datasets. "*" indicates that this method is a training-free approach in test-time adaptation task.

Method	ImageNet	ImageNet-A	ImageNet-V2	ImageNet-R	ImageNet-S	*Average*	*OOD Average*
CLIP-ResNet-50	59.81	23.24	52.91	60.72	35.48	46.43	43.09
CoOp	**63.33**	23.06	55.40	56.60	34.67	46.61	42.43
CoCoOp	<u>62.81</u>	23.32	55.72	57.74	34.48	46.81	42.82
Tip-Adapter	62.03	23.13	53.97	60.35	35.74	47.04	43.30
TPT	60.74	26.67	54.70	59.11	35.09	47.26	43.89
DiffTPT	60.80	<u>31.06</u>	<u>55.80</u>	58.80	37.10	48.71	45.69
TDA*	61.35	30.29	55.54	<u>62.58</u>	<u>38.12</u>	<u>49.58</u>	<u>46.63</u>
SOBA (Ours)*	61.85	**31.54**	**55.92**	**62.91**	**38.85**	**50.21**	**47.31**
CLIP-ViT-B/16	68.34	49.89	61.88	77.65	48.24	61.20	59.42
CoOp	**71.51**	49.71	64.20	75.21	47.99	61.72	59.28
CoCoOp	<u>71.02</u>	50.63	64.07	76.18	48.75	62.13	59.91
Tip-Adapter	70.75	51.04	63.41	77.76	48.88	62.37	60.27
TPT	68.98	54.77	63.45	77.06	47.94	62.44	60.81
DiffTPT	70.30	55.68	65.10	75.00	46.80	62.28	60.52
MTA	69.29	57.41	63.61	76.92	48.58	63.16	61.63
MTA+Ensemble	70.08	58.06	64.24	78.33	49.61	64.06	62.56
TDA*	69.51	<u>60.11</u>	<u>64.67</u>	<u>80.24</u>	<u>50.54</u>	<u>65.01</u>	<u>63.89</u>
SOBA (Ours)*	70.90	**61.06**	**65.83**	**80.79**	**52.57**	**66.23**	**65.06**

Throughout the process, we set $P = I_n$ and $Q = Q_c$ (Q_c is obtained from Eq. (8)). Since $\hat{w}_{ij}$ is the (i, j)-th element of $\hat{W}$, we only need to multiply the unitary matrix by W to achieve the SOBA mapping. During this time, we estimate the covariance matrix using the following approach:

$$C = \frac{1}{N}\sum_{k=1}^{N} \frac{\sum_{i=1}^{M_k} \mathbb{I}_{\hat{l}=k}(\boldsymbol{f}_{test,i} - \mu_k)(\boldsymbol{f}_{test,i} - \mu_k)^{\mathrm{T}}}{\sum_{i=1}^{M_k} \mathbb{I}_{\hat{l}=k}}, \tag{11}$$

where to reduce the computational burden, we adopt the GDA [30] assumption for calculating the covariance matrix, which states that all classes follow a distribution with a common covariance.

The test sample feature $\boldsymbol{f}_{\text{test}}$ is transformed using Eq. 7 to obtain $\hat{\boldsymbol{f}}_{\text{test}}$. Finally, the SOBA classifier is formulated as follows:

$$logits_{\text{trans}} = \text{Linear}(\hat{\boldsymbol{f}}_{test}, \hat{\mu}). \tag{12}$$

Additionally, during the inference process, we update the covariance and mean every 10% of the test samples to further reduce the computational burden. Ultimately, we employ mixed predictions to consolidate the final logits output. Therefore, the output logits for the test images are calculated as follows:

$$logits = \hat{\boldsymbol{f}}_{test}{\mathbf{W}_t}^{\mathrm{T}} + \alpha \times logits_{\text{trans}}, \tag{13}$$

where α is a hyperparameter.

4 Experiment

4.1 Experimental Setup

Benchmarks. Based on previous work [8, 10, 9, 31], we selected the out-of-distribution (OOD) benchmark and the cross-dataset benchmark as the foundational experiments for our study.

- For the **OOD benchmark**, we test the effectiveness of our method on out-of-distribution datasets using ImageNet and its four OOD sub-datasets, which include ImageNet-A [32], ImageNet-R [33], ImageNet-V2 [34], and ImageNet-S [35]. The purpose of the OOD benchmark is to evaluate the model's generalization ability to data from the same class but different domain distributions.

Table 2: **Results on the Cross-Dataset Benchmark.** Compare the performance of our method with existing methods on Cross-Dataset benchmark. Our method achieves the highest average accuracy on both backbones. The best results are in **bold** and the second-best results are <u>underlined</u>. Among the methods we compared, CoOp [3] and CoCoOp [4] are fine-tuned on the training set; TPT [8], DiffTPT [10] and HisTPT [11] require backpropagation to update the prompts; TDA [9], and our method do not require any backpropagation to update parameters. *Average* refers to the average accuracy across all datasets. "*" indicates that this method is a training-free approach in test-time adaptation task.

Method	Aircraft	Caltech101	Cars	DTD	EuroSAT	Flower102	Food101	Pets	SUN397	UCF101	*Average*
CLIP-ResNet-50	16.11	87.26	55.89	40.37	25.79	62.77	74.82	82.97	60.85	59.48	56.63
CoOp	15.12	86.53	55.32	37.29	26.20	61.55	75.59	87.00	58.15	59.05	56.18
CoCoOp	14.61	87.38	56.22	38.53	28.73	65.57	76.20	<u>88.39</u>	59.61	57.10	57.23
TPT	17.58	87.02	58.46	40.84	28.33	62.69	74.88	84.49	61.46	60.82	57.66
DiffTPT	17.60	86.89	60.71	40.72	41.04	63.53	<u>79.21</u>	83.40	62.72	62.67	59.85
HisTPT	**18.10**	87.20	<u>61.30</u>	41.30	**42.50**	67.60	**81.30**	84.90	<u>63.50</u>	64.10	<u>61.18</u>
TDA*	17.61	<u>89.70</u>	57.78	<u>43.74</u>	<u>42.11</u>	**68.74**	77.75	86.18	62.53	<u>64.18</u>	61.03
SOBA (Ours)*	<u>17.70</u>	**90.18**	**61.40**	**44.80**	41.51	<u>67.61</u>	77.82	**88.69**	**65.65**	**66.77**	**62.20**
CLIP-ViT-B/16	23.22	93.55	66.11	45.04	50.42	66.99	82.86	86.92	65.63	65.16	64.59
CoOp	18.47	93.70	64.51	41.92	46.39	68.71	85.30	89.14	64.15	66.55	63.88
CoCoOp	22.29	93.79	64.90	45.45	39.23	70.85	83.97	<u>90.46</u>	66.89	68.44	64.63
TPT	24.78	94.16	66.87	<u>47.75</u>	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
MTA	25.32	94.13	68.05	45.59	38.71	68.26	84.95	88.22	64.98	68.11	64.63
MTA+Ensemble	25.20	94.21	68.47	45.90	45.36	68.06	85.00	88.24	66.60	68.69	65.58
HisTPT	**26.90**	<u>94.50</u>	<u>69.20</u>	**48.90**	49.70	71.20	**89.30**	89.10	67.20	70.10	<u>67.61</u>
TDA*	23.91	<u>94.24</u>	67.28	47.40	<u>58.00</u>	<u>71.42</u>	86.14	88.63	<u>67.62</u>	<u>70.66</u>	67.53
SOBA (Ours)*	<u>25.62</u>	**94.60**	**71.12**	46.87	**59.44**	**71.66**	<u>86.69</u>	**92.48**	**70.63**	**74.12**	**69.32**

- For the **cross-dataset benchmark**, we use 10 public datasets to evaluate the cross-dataset classification capability of our method. Each dataset comes from different classes and domains, including: Aircraft [36], Caltech101 [37], Car [38], DTD [39], EuroSAT [40], Flowers102 [41], Food101 [42], Pets [43], SUN397 [44], and UCF101 [45].

Comparison Methods. We compare our method with zero-shot CLIP [1], CoOp [3], CoCoOp [4], Tip-Adapter [6], and other state-of-the-art (SOTA) methods in the TTA domain that do not require a training set, including TPT [8], DiffTPT [10], MTA [31], HisTPT [11], and TDA [9]. Among these, Tip-Adapter cannot be evaluated on the cross-dataset benchmark because it is unable to handle unseen classes during the testing phase. Additionally, we do not include MTA in the comparison for experiments with ResNet-50 as the backbone, as there is no data available for MTA on ResNet-50. Furthermore, MTA+Ensemble refers to the ensemble prediction method provided in the MTA paper. And HisTPT does not include experiments on the OOD benchmark. Lastly, as the HisTPT paper does not include experiments on the OOD benchmark, we do not compare with it in this setting. Notably, the decision boundary of TDA is based on the original CLIP's feature space, while our method transcends this space.

Implementation Details. Our method is built upon pre-trained CLIP [1], where the text encoder of CLIP is a Transformer [46], and the image encoder can be either ResNet [47] or Vision Transformer [48]. Since our method is training-free, all text prompts are manually crafted. To construct the dynamic queue, we set the batch size to 1. For the OOD benchmark, we conduct a hyperparameter search on ImageNet and apply the resulting hyperparameters to the remaining four OOD datasets. For the cross-dataset benchmark, we conducted experiments with various queue lengths and ultimately set the queue length to 16. For detailed experimental results, please refer to Appendix C. Additionally, we use top-1 accuracy as the evaluation metric for our experiments, and all experiments are performed on an NVIDIA Quadro RTX 6000 GPU.

4.2 Comparison with State-of-the-arts

We compare our method against zero-shot CLIP, CoOp, CoCoOp, Tip-Adapter, TPT, DiffTPT, HisTPT, MTA, and TDA. Notably, CoOp, CoCoOp, and Tip-Adapter require a training set for optimization, while TPT, DiffTPT, MTA, TDA, and our method do not. Like TPT, DiffTPT, MTA, and TDA, we evaluate our method on both the **OOD benchmark** and the **cross-dataset benchmark**.

Table 3: **Performance improvement of our method over cache baseline on both benchmarks.** The experiments employ ViT-B/16 as the backbone. Compared to the baseline, our method exhibits improved performance across all datasets.

(a) Performance improvement of our method over cache baseline on OOD benchmark.

Method	ImageNet	ImageNet-A	ImageNet-V2	ImageNet-R	ImageNet-S	*Average*	*OOD Average*
Baseline	69.04	60.04	64.54	80.16	49.39	64.63	63.53
+SOBA (Ours)	**70.90**	**61.06**	**65.83**	**80.79**	**52.57**	**66.23**	**65.06**
Improvement	**+1.86**	**+1.02**	**+1.29**	**+0.63**	**+3.18**	**+1.60**	**+1.53**

(b) Performance improvement of our method over cache baseline on Cross-Dataset benchmark.

Method	Aircraft	Caltech101	Cars	DTD	EuroSAT	Flower102	Food101	Pets	SUN397	UCF101	*Average*
Baseline	24.72	94.07	67.79	45.80	55.06	71.15	86.4	88.41	67.69	70.24	67.13
+SOBA (Ours)	**25.62**	**94.60**	**71.12**	**46.87**	**59.44**	**71.66**	**86.69**	**92.48**	**70.63**	**74.12**	**69.32**
Improvement	**+0.90**	**+0.53**	**+3.33**	**+1.07**	**+4.38**	**+0.51**	**+0.29**	**+4.07**	**+2.94**	**+3.88**	**+2.19**

Table 4: Comparisons of our method with CLIP-ResNet-50, TPT, DiffTPT and TDA in terms of efficiency and accuracy. The experiments are conducted on ImageNet.

Method	Training-free	Testing Time	Accuracy	Improved
CLIP-ResNet-50	✓	**12min**	59.81	0.
TPT	✗	12h 50min	60.74	0.93
DiffTPT	✗	34h 45min	60.80	0.99
TDA	✓	16min	61.35	1.54
SOBA (Ours)	✓	13min 46s	**61.85**	**2.04**

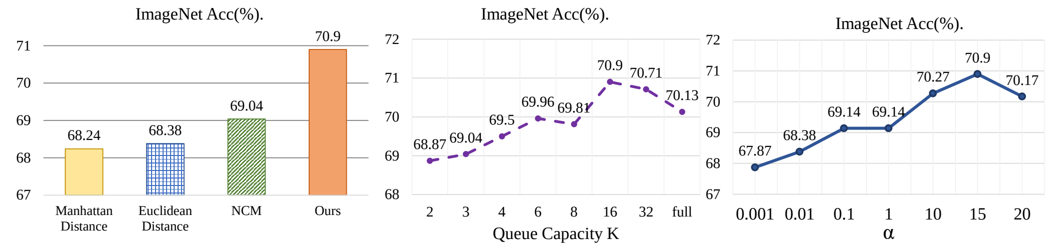
Results on the Out-of-Distribution Benchmark. Table 1 provides a comparison between our method and state-of-the-art (SOTA) approaches across different backbones on ImageNet and four out-of-distribution (OOD) datasets. Our method surpasses existing approaches on all OOD datasets. Notably, it outperforms TDA with an increase of 0.68% in OOD average accuracy using the ResNet-50 backbone and **1.17%** with the ViT-B/16 backbone. Additionally, our approach demonstrates a significant **3.43%** improvement over MTA with the ViT-B/16 backbone. These results affirm the effectiveness of exploring new decision boundaries beyond the original CLIP decision surface, validating our approach.

Efficiency Comparison. As shown in Table 4, to assess the efficiency of our method using ResNet-50 as the backbone, we compared it with three existing test-time adaptation methods on the ImageNet dataset, focusing on inference speed and accuracy. The performance metrics for CLIP-ResNet-50, TPT, DiffTPT, and TDA are sourced from the TDA paper. While our method sacrifices slight efficiency compared to zero-shot CLIP, it achieves a 2.04% accuracy improvement. Unlike TPT and DiffTPT, which require backpropagation, our method significantly outperforms them in efficiency. Compared to TDA, our method enhances both efficiency and accuracy, improving inference time by 2m 14s and accuracy by 0.5%. These results demonstrate the efficiency and suitability of our approach for test-time adaptation.

Results on the Cross-Datasets Benchmark. To further validate the feasibility and effectiveness of our approach, we conducted comparisons with SOTA methods across 10 datasets spanning diverse categories and domains. As shown in Table 2, our method consistently outperforms competitors on both backbones tested. Using ResNet-50, our approach achieved top performance on 6 out of 10 datasets, with an average accuracy improvement of **1.13%** over TDA. With ViT-B/16, our method led on 7 out of 10 datasets, surpassing TDA with a **1.79%** increase in average accuracy. The performance on the cross-dataset benchmark further demonstrates that our method remains effective even when faced with datasets from different classes and domains. Moreover, our method does not require additional training or backpropagation on both benchmarks, making it well-suited for testing adaptation tasks with CLIP.

4.3 Ablation Studies

In this section, we conduct ablation experiments to analyze the effectiveness of our design. Our baseline method is the one mentioned in Section 3.1.



(a) Comparison with Other Classifiers (b) The Impact of Queue Capacity (c) The Impact of Hyperparameter α

Figure 3: Subfigure (a) shows a comparison with other classifiers, where our SOBA achieves the best performance. Subfigure (b) presents a study on different dynamic queue lengths. Subfigure (c) presents a study on the impact of the hyperparameter α.

Effectiveness of SOBA. To clearly illustrate the effectiveness of our method, we compare it with a simple yet effective baseline. In Table 3, we report the ablation experiments on the OOD benchmark and cross-dataset benchmark, respectively. Since the baseline method also does not involve backpropagation and is based on the original CLIP feature space, comparing it with this baseline allows us to directly observe the pure benefit of the space rotation provided by SOBA.

Compared to baseline, our work demonstrates significant improvements across nearly all datasets in both benchmarks. Compared to the baseline, on the OOD benchmark, our two evaluation metrics, *average* and *OOD average*, improved by **1.6%** and **1.53%**. On the cross-dataset benchmark, we achieved a **2.19%** improvement in *average*. Combining our finding with the comparisons to TDA in Section 4.2, that rely on the original CLIP feature space, we can conclude that applying a basis transformation to rotate the original space is a feasible solution to address the TTA problem, and it achieves better performance than the original CLIP feature space.

Comparison with Other Classifiers. In Fig. 3(a), we present a comparison of our method with other classifiers. Due to changes in the feature space, directly minimizing the Manhattan (L1) distance and Euclidean (L2) distance to class centers is no longer applicable, and it even results in degradation compared to zero-shot CLIP. Our method, compared to the basic NCM classifier, achieves better decision boundaries by utilizing the rotated space, further addressing the test-time adaptation problem.

Hyperparameter Sensitivity Analysis.

- **Queue Capacity K.** In Fig. 3(b), we report the impact of dynamic queue Capacity. We find that as the Capacity of the dynamic queue increases, the overall accuracy shows a trend of first increasing and then decreasing. This can be understood as follows: when the queue Capacity is small, the stored features are very representative, but as the queue Capacity increases, some easily confusable features are added, affecting subsequent judgments. In this paper, we select 16 as the storage limit for each class in our dynamic queue on the OOD benchmark. For the ablation study on queue length in the Cross-Dataset benchmark, please refer to the Appendix C.2.

- **Hyperparameter α.** In Fig. 3(c), we illustrate the impact of α from Eq. (13). Based on the performance on ImageNet, we ultimately select $\alpha = 15$ as the final value. For the effect of α on other datasets, please refer to the Appendix C.2.

5 Conclusion

In this work, we introduce a space rotation with basis transformation (SOBA) method, designed to overcome the limitations of the training-free TTA paradigm in the feature space. By leveraging SOBA, we perform a rotation and reconstruction of the original feature space, thereby tackling the adaptation issues that arise from distribution changes during testing. Experimental results across various benchmarks have demonstrated that our method not only outperforms state-of-the-art approaches but is also easy to implement and highly efficient. Detection and semantic segmentation tasks can still be regarded as fine-grained classification tasks. In future work, we plan to extend the application of SOBA to related domains to validate its effectiveness across various visual tasks.

References

[1] 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 *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[2] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021.

[3] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

[4] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16816–16825, 2022.

[5] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19113–19122, 2023.

[6] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European conference on computer vision*, pages 493–510. Springer, 2022.

[7] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, 132(2):581–595, 2024.

[8] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. *Advances in Neural Information Processing Systems*, 35:14274–14289, 2022.

[9] Adilbek Karmanov, Dayan Guan, Shijian Lu, Abdulmotaleb El Saddik, and Eric Xing. Efficient test-time adaptation of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14162–14171, 2024.

[10] Chun-Mei Feng, Kai Yu, Yong Liu, Salman Khan, and Wangmeng Zuo. Diverse data augmentation with diffusions for effective test-time prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2704–2714, 2023.

[11] Jingyi Zhang, Jiaxing Huang, Xiaoqin Zhang, Ling Shao, and Shijian Lu. Historical test-time prompt tuning for vision foundation models.

[12] Corinna Cortes. Support-vector networks. *Machine Learning*, 1995.

[13] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[14] Gilbert Strang. Linear algebra and its applications, 2000.

[15] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

[16] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training. *arXiv preprint arXiv:2111.07783*, 2021.

[17] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11583–11592, 2022.

[18] Muhammad Maaz, Hanoona Rasheed, Salman Khan, Fahad Shahbaz Khan, Rao Muhammad Anwer, and Ming-Hsuan Yang. Class-agnostic object detection with multi-modal transformer. In *European conference on computer vision*, pages 512–531. Springer, 2022.

[19] Shaokun Wang, Yifan Yu, Yuhang He, and Yihong Gong. Enhancing pre-trained vits for downstream task adaptation: A locality-aware prompt learning method. In *ACM Multimedia 2024*, 2024.

[20] Zhengbo Wang, Jian Liang, Lijun Sheng, Ran He, Zilei Wang, and Tieniu Tan. A hard-to-beat baseline for training-free clip-based adaptation. *arXiv preprint arXiv:2402.04087*, 2024.

[21] Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free online test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8344–8353, 2022.

[22] Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *Advances in neural information processing systems*, 35:38629–38642, 2022.

[23] Longhui Yuan, Binhui Xie, and Shuang Li. Robust test-time adaptation in dynamic scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15922–15932, 2023.

[24] Jian Zhang, Lei Qi, Yinghuan Shi, and Yang Gao. Domainadaptor: A novel approach to test-time adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18971–18981, 2023.

[25] 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.

[26] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[27] Taolin Zhang, Jinpeng Wang, Hang Guo, Tao Dai, Bin Chen, and Shu-Tao Xia. Boostadapter: Improving vision-language test-time adaptation via regional bootstrapping.

[28] Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2020.

[29] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2624–2637, 2013.

[30] Trevor Hastie and Robert Tibshirani. Discriminant analysis by gaussian mixtures. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):155–176, 1996.

[31] Maxime Zanella and Ismail Ben Ayed. On the test-time zero-shot generalization of vision-language models: Do we really need prompt learning? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23783–23793, 2024.

[32] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15262–15271, 2021.

[33] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021.

[34] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019.

[35] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, 32, 2019.

[36] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[37] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004.

[38] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[39] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014.

[40] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

[41] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008.

[42] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pages 446–461. Springer, 2014.

[43] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

[44] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010.

[45] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. A dataset of 101 human action classes from videos in the wild. *Center for Research in Computer Vision*, 2(11):1–7, 2012.

[46] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[48] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[49] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction have already stated that our work leverages the inherent differences between categories. We reconstruct the original feature space using a basis transformation technique to reveal clearer decision boundaries.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have outlined the limitations of our method in the Limitation section of the appendix: detection and semantic segmentation tasks can still be regarded as fine-grained classification problems. In future work, we plan to apply SOBA to these related domains to evaluate its effectiveness across different vision tasks.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We provide a detailed derivation and justification of the basis transformation in the Method section of the paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: All necessary implementation details, including datasets, hyperparameters, training settings, and evaluation protocols, are provided in the main text and appendix. Pseudocode for key algorithms is also included.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All datasets used are publicly available, and the code will be released shortly.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide detailed descriptions of all training and testing procedures in the Experiments section, and the corresponding results can be found in both the main text and the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The experimental error stems from the randomness of the test sample stream. We report the average results over three runs with different random test sample orders, as stated in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The detailed computational cost can be found in Table 4, and a thorough comparative analysis is provided in the Experiments section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in this paper fully complies with the NeurIPS Code of Ethics in all aspects.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

Justification: The research presented in this paper is unrelated to malicious uses (such as content forgery or surveillance) and fairness issues (such as gender or racial bias).

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper only involves the use of publicly available pre-trained models and does not propose any new pre-trained models.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets and assets used are properly cited and comply with their respective licenses.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not introduce new datasets or models that require additional documentation.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or human subject research.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or human subject research.

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: No large language models were used as part of the core methodology. If LLMs were used for minor language editing, it does not affect the scientific content of the paper.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Limitations

Detection and semantic segmentation tasks can still be regarded as fine-grained classification tasks. In future work, we plan to extend the application of SOBA to related domains to validate its effectiveness across various visual tasks.

B Additional Implementation of SOBA

In this section, we provide a detailed description of the overall process of handling the feature space with basis vectors in our SOBA method.

B.1 SOBA Process

The SOBA process includes the following key steps: for each test sample x_{test}, the algorithm first extracts the image feature f_{test} and text features W_t using CLIP's visual encoder $E_v(\theta_v)$ and text encoder $E_v(\theta_v)$, and calculates the original CLIP logits by Eq. 1. It then generates pseudo-labels by applying one-hot encoding to the original logits by Eq. 2, and updates the dynamic queue, which stores the image features, pseudo-labels, and logits. After that, we compute the prototype for each pseudo-class and calculates the covariance matrix of the queue by Eq. 9 and Eq. 11.

Next, the prototypes are rotated using the SOBA method to obtain new class prototypes by Eq. 10, and the transformed logits are computed based on these rotated prototypes by Eq. 12. Finally, the algorithm combines the original logits and the transformed logits with a weighting factor α to produce the final prediction. It is worth noting that to ensure the stability and accuracy of the obtained orthogonal basis and class prototypes, we update the prototypes every 10% of the test samples. This strategy allows the algorithm to optimize the model's adaptability while maintaining computational efficiency, and reduces the impact of bases constructed from too few samples on the final results. The overall process is presented in Algorithm 1.

B.2 Queue Update Process

In this section, we explain how to perform enqueue and dequeue operations on the queue.

First, for each test feature x_{test}, the algorithm checks whether the queue $L_{\hat{l}}^{t-1}$ corresponding to the current pseudo-label $\hat{l}$ is full. If the queue is not full, the current feature f_{test} and its corresponding pseudo-label $\hat{l}$ are simply enqueued, generating a new queue L^t. If the queue is full, the algorithm first calculates the maximum entropy H_{max} in the queue, which represents the average uncertainty of the current features. Then, the algorithm compares the entropy of the current feature's logits $H(logits_{ori})$ with the maximum entropy H_{max}. If the current feature's entropy is smaller than the maximum entropy, it indicates that the feature is more certain, and the algorithm removes the feature with the highest entropy from the queue and enqueues the current feature; otherwise, the queue remains unchanged. Finally, the algorithm returns the updated queue L^t, which helps manage the updates of features and pseudo-labels, ensuring that the queue adapts to new data over time. The overall process is presented in Algorithm 2.

C Additional Ablation Study

C.1 Additional Robustness Analysis

Table 5: **Analysis of different pseudo-label noise ratios.** The experiments are conducted on ImageNet.

Noise Ratio	Accuracy	Improved over CLIP
SOBA w 60%	61.29	1.48
SOBA w 40%	61.73	1.92
SOBA w 20%	61.79	1.98
SOBA	**61.85**	**2.04**

Stability and Impact of Noise. To verify the robustness of SOBA to noisy pseudo-labels, we introduced Gaussian white noise into the pseudo-labels for testing, as shown in Table 5. Under low noise ratios (20% and 40%), SOBA demonstrates strong capability in correcting the decision boundaries. When the noise ratio increases (exceeding 50%), SOBA still outperforms the original CLIP. Overall, SOBA maintains robustness in constructing clear decision boundaries under noisy conditions.

Table 6: **Performance on High-Entropy Datasets.** “Average" represents the mean performance on the Cross-Dataset benchmark.

Method	UCF101	Cars	Aircraft	Average
CLIP	65.16	66.11	23.22	64.59
TDA	70.66	67.28	23.91	67.53
SOBA	**74.12**	**71.12**	**25.62**	**69.32**

Feasibility of the shared Gaussian assumption. Table 6 shows SOBA's strong performance on datasets with **significant domain shifts**, demonstrating that the GDA assumption is valid even under severe domain shifts.

Table 7: **Results on the OOD Benchmark**. Compare the performance of our method with Boost-Adapter [27] on OOD benchmark. SOBA$^+$ represents the performance of our method based on the settings of BoostAdapter, adopting the same data augmentation strategy. The backbone is CLIP ViT-B/16.

Method	ImageNet-A	ImageNet-V2	ImageNet-R	ImageNet-S	*OOD Average*
BoostAdapter [27]	**64.53**	65.51	80.95	51.28	65.57
SOBA (**Ours**)	61.06	65.83	80.79	52.57	65.06
SOBA$^+$ (**Ours**)	63.27	**66.08**	**81.35**	**53.06**	**65.94**

Table 8: **Results on the Cross-Dataset Benchmark.** Compare the performance of our method with BoostAdapter [27] on Cross-Dataset benchmark. SOBA$^+$ represents the performance of our method based on the settings of BoostAdapter, adopting the same data augmentation strategy. The backbone is CLIP ViT-B/16.

Method	Aircraft	Caltech101	Cars	DTD	EuroSAT	Flower102	Food101	Pets	SUN397	UCF101	*Average*
BoostAdapter [27]	27.45	94.77	69.30	45.69	61.22	71.66	87.17	89.51	68.09	71.93	68.68
SOBA (**Ours**)	25.62	94.60	71.12	46.87	59.44	71.66	86.69	92.48	70.63	74.12	69.32
SOBA$^+$ (**Ours**)	**28.07**	**94.82**	**71.49**	**47.24**	**61.90**	**71.93**	**87.52**	**92.86**	**71.11**	**74.28**	**70.12**

Discussion on the presence of high-entropy or ambiguous images in the queue. Our queue follows consistent update rules across datasets, even in high-entropy scenarios (details are provided in Appendix B). For example, as shown in Table 6, on the highly entropic Aircraft dataset, SOBA still achieves improvements similar to the overall average over TDA, demonstrating its robustness.

C.2 Additional Hyperparameter Aensitivity Analysis

This section supplements the ablation experiment on queue capacity and hyperparameter α.

Queue Capacity K. For the Pets dataset [43], the best accuracy is achieved when the queue capacity per class is 32. We believe the reason is that the differences between different classes in the Pets dataset are significant, as these classes not only exhibit distinct visual features (such as fur color, shape, and body size), but also show considerable diversity in terms of image background, posture, and camera angle. Therefore, increasing the queue capacity can better capture the information of the feature space, allowing the reconstructed basis and class prototypes to more effectively reflect the differences between classes. Finally, we used $K = 16$ as the overall queue capacity for the cross-dataset benchmark.

Hyperparameter α. For hyperparameter α, as shown in the Fig. 4, we performed a hyperparameter search on ImageNet and set α to 15 for all benchmarks (although some datasets may have better settings, we unified α to 15 for consistency).

Table 9: **Results on the Cross-Dataset Benchmark.** The performance of SOBA with different K on the Cross-Dataset benchmark. Due to the complexity of the datasets in the cross-dataset benchmark, the performance of each dataset may vary differently as the queue capacity increases. The backbone used in the experiments is ViT-B.

K	Aircraft	Caltech101	Cars	DTD	EuroSAT	Flower102	Food101	Pets	SUN397	UCF101	*Average*
2	24.72	93.59	67.79	45.80	55.06	71.34	86.40	91.61	67.79	73.09	67.72
4	24.99	93.91	68.90	45.33	54.30	71.50	86.59	91.63	68.15	72.40	67.77
6	25.08	94.24	70.26	45.39	54.63	71.54	86.69	91.28	69.30	72.93	68.13
8	25.32	**94.60**	70.17	45.98	58.79	**71.68**	86.57	91.77	69.41	73.83	68.81
16	**25.62**	**94.60**	71.12	**46.87**	**59.44**	71.66	86.69	92.48	**70.63**	**74.12**	**69.32**
32	25.27	93.31	**71.22**	46.34	58.28	71.38	**86.79**	**92.80**	69.35	72.77	68.75
full	25.21	93.31	70.64	45.81	58.11	71.38	86.53	92.74	68.91	72.55	68.52

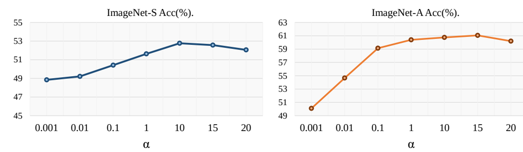


Figure 4: The Impact of Hyperparameter α.

D Additional Experiment Results

In this section, we present a comparison between our method and BoostAdapter. To ensure fairness, we adopt the same experimental settings as BoostAdapter, which allow storing augmented samples and applying data augmentation in the Cross-Dataset benchmark setup—two key differences from previous TTA methods.

Tables 7 and 8 present a comparison between our method and BoostAdapter on the OOD benchmark and Cross-Dataset benchmark, respectively. Under the same experimental settings, our method (SOBA^{+}) achieves superior performance, demonstrating that the reconstructed feature space in our approach exhibits better separability compared to the original CLIP feature space.

E Additional Experimental Details

E.1 Additional Benchmark Details

In this section, we provide detailed information on the two benchmarks used in our work.

OOD Benchmark. OOD benchmark is used to validate the model's ability to generalize to data of the same class but with different styles, assessing its robustness and effectiveness against distributional shifts. For the OOD benchmark, we used ImageNet [49] along with four OOD sub-datasets to evaluate our method's performance on out-of-distribution data. These four datasets include ImageNet-A [32], ImageNet-R [33], ImageNet-V2 [34], and ImageNet-S [35]. Below, we provide a brief overview of each OOD dataset.

- **ImageNet-A** [32]: ImageNet-A is a curated dataset containing 200 challenging classes of images for standard ImageNet-trained models. The dataset is composed of images from the real world that are likely to cause model misclassification, specifically selected to highlight the limitations of traditional models when recognizing out-of-distribution or adversarial samples.
- **ImageNet-R** [33]: ImageNet-R is a dataset derived from ImageNet, specifically designed to test model robustness under significant changes in visual style, covering 200 classes. "R" stands for "Renditions," and the dataset includes images in a variety of artistic styles, such as paintings, cartoons, and sculptures. These images differ significantly from standard ImageNet photographs, making them particularly suitable for evaluating a model's ability to generalize beyond typical photographic representations.
- **ImageNet-V2** [34]: ImageNet-V2 is a dataset designed to evaluate the consistency and robustness of models trained on the original ImageNet dataset, consisting of 1000 classes. It was created by re-sampling the original ImageNet categories using methods that are

similar but not identical to the original collection process. ImageNet-V2 aims to measure the generalization ability of models, as it mimics the distribution of the original dataset while incorporating new, previously unseen samples.

- **ImageNet-S** [35]: ImageNet-S is a dataset derived from ImageNet, containing 1000 classes, specifically designed to evaluate a model's sensitivity to background changes and its ability to focus on salient features. "S" stands for "Sketches," and the dataset consists of black-and-white sketches of the original ImageNet classes. The simplified and abstract nature of the sketches challenges models to classify images based solely on basic contours and shapes, rather than relying on background context or texture information.

Cross-Dataset Benchmark. The cross-dataset benchmark consists of 10 image classification datasets, each representing a distinct domain and category, designed to evaluate the model's effectiveness and generalization capability across diverse scenarios. The benchmark includes the following datasets: Caltech101 for general image classification; OxfordPets (Pets), StanfordCars (Cars), Flowers102, Food101, and FGVCAircraft (Aircraft) for fine-grained image classification; EuroSAT for satellite imagery classification; UCF101 for action recognition; DTD for texture classification; and SUN397 for scene classification.

For the number of classes and the number of test samples for each dataset in both benchmarks, please refer to the table 10.

Table 10: Datasets Information.

Dataset	Classes	Test Samples
OOD benchmark		
ImageNet	1,000	50,000
ImageNet-V2	1,000	10,000
ImageNet-S	1,000	50,000
ImageNet-A	200	7,500
ImageNet-R	200	30,000
Cross-Dataset benchmark		
Aircraft	100	3,333
Caltech101	101	2,465
Cars	196	8,041
DTD	47	1,692
EuroSAT	10	8,100
Flowers102	102	2,463
Food101	101	30,300
Pets	37	3,669
SUN397	397	19,850
UCF101	101	3,783

E.2 Additional Comparison Methods Details

In this section, we provide a detailed description of the methods compared in our work.

CoOp [3]: CoOp [3] aims to perform automatic prompt optimization for vision-language models (e.g., CLIP) to achieve better few-shot learning and cross-domain generalization. CoOp replaces manually crafted prompt tokens with learnable context vectors while keeping the pre-trained model parameters unchanged. These context vectors are optimized by learning task-specific information from the data, significantly improving model performance.

CoCoOp [4]: CoCoOp [4] is an extension of the previous CoOp method. CoCoOp learns a lightweight neural network to generate context prompts conditioned on the input image, making the prompts dynamic rather than static, and adjusting them for each instance. This allows CoCoOp to better adapt to class variations, thereby enhancing the model's generalization ability to new classes.

Tip-Adapter [6]: Tip-Adapter [6] is designed to adapt the CLIP model for few-shot classification in a training-free manner. Tip-Adapter is based on a key-value cache model, constructing a non-parametric adapter from a small number of training samples without any additional training. It extracts features from few-shot images using CLIP's visual encoder and stores these features along with corresponding pseudo-labels in a cache, leveraging feature retrieval for inference. This approach enables the CLIP model to incorporate few-shot knowledge without retraining, achieving performance comparable to models that require training.

TPT [8]: TPT [8] dynamically adjusts adaptive prompts during testing, using only a single test sample without requiring additional training data or annotations. The method optimizes prompts by minimizing the marginal entropy between augmented views to ensure consistent predictions for different augmented versions of each test sample. Additionally, TPT introduces a confidence selection mechanism to filter out low-confidence augmented samples, thereby reducing the impact of noise.

DiffPT [10]: DiffTPT [10] utilizes a pre-trained diffusion model to generate diverse and informative augmented data, while maintaining prediction accuracy through cosine similarity filtering. This method combines traditional data augmentation with diffusion-based augmentation, enabling the model to improve its adaptability when encountering novel data without the need for retraining.

MTA [31]: MTA [31] employs a robust multimodal MeanShift algorithm to manage augmented views during testing by directly optimizing the quality evaluation of augmented views, referred to as the "inherence score." This method does not require prompt tuning and does not rely on complex training processes, enabling efficient adaptation to new data.

TDA [9]: TDA [9] uses a lightweight key-value cache to dynamically maintain a small number of pseudo-labels and test sample features. It gradually adapts to test data through progressive pseudo-label refinement, without requiring backpropagation, making it highly efficient. TDA also introduces a negative pseudo-label mechanism, which assigns pseudo-labels to certain negative classes to reduce the impact of noisy pseudo-labels. By combining both positive and negative caches, TDA significantly improves the model's classification accuracy and generalization ability without retraining, while also greatly reducing test time.

Algorithm 1 The testing loop of proposed **SOBA** method for test-time adaptation

Input: CLIP visual encoder $E_v(\theta_v)$, text encoder $E_t(\theta_t)$, testing dataset D_{test}, number of classes N, N text descriptions T of N classes, original basis $\mathcal{E}$, dynamic queue L, hyper-parameter α, queue capacity K.
for each test sample x_{test} in D_{test} **do**
    Image embedding: $f_{test} \leftarrow E_v(\theta_v, x_{test})$
    Text embeddings: $W_t \leftarrow E_t(\theta_t, T)$
    CLIP logits: $logits_{ori} \leftarrow f_{test} W_t^{\mathrm{T}}$
    Pseudo-label of x_{test}: $\hat{l} \leftarrow \mathtt{OneHot}(logits_{ori})$
    $L \leftarrow \mathtt{Update}(L, f_{test}, \hat{l}, logits_{ori})$ ▷ See Algorithm 2
    for each pseudo-class $\hat{l}_k$ in L **do**
        Get prototype of class $\hat{l}_k$: $\mu_k \leftarrow \frac{\sum_{i=1}^{M_k} \mathbb{I}_{\hat{l}=k} f_{test,i}}{\sum_{i=1}^{M_k} \mathbb{I}_{\hat{l}=k}}$
    end for
    Get covariance C of L: $C \leftarrow \frac{1}{N} \sum_{k=1}^{N} \frac{\sum_{i=1}^{M_k} \mathbb{I}_{\hat{l}=k} (f_{test,i} - \mu_k)(f_{test,i} - \mu_k)^{\mathrm{T}}}{\sum_{i=1}^{M_k} \mathbb{I}_{\hat{l}=k}}$
    Space rotation: $\hat{\mu} \leftarrow \mathtt{SOBA}(\mu, C)$, $\hat{f}_{test} \leftarrow \mathtt{SOBA}(f_{test}, C)$ ▷ See Equation (7) and (10)
    SOBA logits: $logits_{trans} \leftarrow \mathtt{Linear}(\hat{f}_{test}, \hat{\mu})$
    Final inference: $logits \leftarrow logits_{ori} + \alpha \times logits_{\mathrm{trans}}$
end for
return $logits$ ▷ return prediction based on the mode

Algorithm 2 Queue update process

```
Input: CLIP logits of f_test: logits_ori, image embedding: f_test, pseudo-label of f_test: l̂, old queue: L^{t-1}, queue capacity: K.
if |L_l̂^{t-1}| < K then
    L_l̂^t ← EnQueue(f_test, L_l̂^{t-1})
else
    H_max ← max(H(L_l̂^{t-1}))                          ▷ Get the maximum entropy in L_l̂^{t-1}.
    if H(logits_ori) < H_max then
        Dequeue feature with H_max: L_l̂^{t-1} ← DeQueue(f_test^ent, L_l̂^{t-1})
        Enqueue feature f_test: L_l̂^t ← EnQueue(f_test, L_l̂^{t-1})
    else
        L_l̂^t ← L_l̂^{t-1}
    end if
end if
return L^t                                             ▷ update the queue
```