# A KERNEL PERSPECTIVE ON FEW-SHOT ADAPTATION OF LARGE VISION-LANGUAGE MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

The growing popularity of Contrastive Language-Image Pretraining (CLIP) has led to its widespread application in various visual downstream tasks. To enhance CLIP's effectiveness, efficient few-shot adaptation techniques have been widely adopted. Among these approaches, training-free methods, particularly caching methods exemplified by Tip-Adapter, have gained attention for their lightweight adaptation without the need for additional fine-tuning. In this paper, we revisit Tip-Adapter from a kernel perspective, showing that caching methods function as local adapters and are connected to a well-established kernel literature. Leveraging this insight, we offer a theoretical understanding of how these methods operate and suggest multiple avenues for enhancing over the Tip-Adapter baseline. Notably, our analysis shows the importance of incorporating global information in local adapters. Therefore, we subsequently propose a global method that learns a proximal regularizer in a reproducing kernel Hilbert space (RKHS) using CLIP as a base learner. Our method, that we call ProKeR (Proximal Kernel ridge Regression), has a closed form solution and achieves state-of-the-art performance across 11 datasets in the standard few-shot adaptation benchmark.

## 1 INTRODUCTION

Large scale vision-language models (VLMs) trained with contrastive learning have gained an increasing attraction in the recent years [41; 21]. These models have shown remarkable performance across a wide range of tasks such as classification [41], segmentation [30] and video understanding [29]. CLIP is one of the most established VLMs [41] offering remarkable zero-shot capabilities in various downstream tasks. It operates by using only the class label within a textual prompt such as "A photo of a {CLASS}" where {CLASS} is the groundtruth text label for each class. However, it is well-known that the zero-shot performance of CLIP is limited in scenarios with large domain shift from the pre-training distribution [10; 34]. To further improve CLIP's generalization, multiple follow-up works proposed to include few-shot data [66; 12; 62] which has shown remarkable performance gains compared to zero-shot CLIP.

Few-shot adaptation of CLIP can be categorized into two types of methods based on whether they require fine-tuning on the few-shot samples or not. Among fine-tuning based methods, prompt learning involves learning continuous tokens instead of hand-crafted templates in CLIP as proposed by CoOp [66] and CoCoOp [65]. Additionally, adapter-based fine-tuning methods operate in the features space to train their classifiers [12; 28]. Despite their promising performance on downstream tasks, fine-tuning methods require additional training costs to learn the new learnable parameters. Tip-Adapter proposed a training-free few-shot adaptation alternative [62]. Using a caching mechanism, Tip-Adapter captures knowledge from the few-shot samples without additional fine-tuning and ensembles it with zero-shot CLIP. This cache model has shown significant improvement over zero-shot performance which has led to follow-up works exemplified by APE [67] and Tip-X [53].

In order to understand caching effectiveness and limitations, we undertake a theoretical analysis to explore the nature of Tip-Adapter. We begin by demonstrating that the adaptation term of Tip-Adapter is a modified version of the well-known Nadaraya-Waston (NW) estimator [33], a local nonparametric regression method that allows Tip-Adapter to effectively capture various distributions. However, the NW estimator is also known to be strongly biased [25; 36]. To mitigate this bias, we leverage existing tools from the literature of kernel methods. An effective extension to NW estimator is locally

linear regression [46]. By fitting a local linear regression around each test point using a closed-form solution, we significantly improve the performance of Tip-Adapter. However, the parameters are estimated locally, which is prone to overfitting in high-dimensional problems [36; 16].

Our analysis shows that caching methods can be understood as local nonparametric regression methods regularised through CLIP pointwise zero-shot predictions. However, this regularization only acts locally and doesn't provide any global information about the few-shot task. Conversely, recent training-based methods rely on global regularizers to incorporate global information [50]. Hence, we ask ourselves the question, *how can we leverage global regularizers for few-shot adapters while conserving the benefits of training-free methods?* In order to design such a global regularizer, we devise two important design choices. Firstly, we restrain the hypothesis space of the learned function to be a reproducing kernel Hilbert space (RKHS). Secondly, using the RKHS norm, we introduce a proximal regularization term to ensure that the obtained solution is close to the base predictor i.e. $f_{\text{clip}}$. Thanks to the properties of the RKHS, minimizing the difference between two functions using the RKHS norm ensures that they are close pointwise. Our method ProKeR provides a more effective way to preserve prior knowledge from the zero-shot predictor and maintains the expressive capacity of the learned functions. Through extensive experiments, we show the effectiveness of our method ProKeR which achieves consistent gains over state-of-the-art training-free methods on standard few-shot classification benchmarks [62] with an absolute average improvement of $3.94\%$ accuracy. In addition, we highlight the robustness and generalizability of the proposed method across different architectures and on out-of-distribution datasets.

**Summary of contributions:**

1. We frame the cache model of Tip-Adapter as a Nadaraya-Waston estimator, a classical kernel regression method and provide a theoretical understanding of how Tip-Adapter operates.
2. Under this new perspective, we propose multiple improvements for Tip-Adapter either using a closed-form local linear regression fit or by incorporating global information.
3. We propose ProKeR, a training-free method that leverages global regularization in a RKHS. Through extensive experiments, ProKeR outperforms existing methods and sets a new state-of-the-art on standard few-shot classification benchmarks.

## 2 Related Work

### 2.1 Vision-Language Pre-trained Models

In recent years, visual-language models (VLMs) have gained an increasing popularity. These methods, exemplified by CLIP [41], DeCLIP [27] and ALIGN [21], employ a contrastive learning framework to learn a vision encoder and a language encoder with a shared representation space between text and images. Trained on large-scale datasts of image-text pairs, VLMs have shown remarkable zero-shot capabilities on downstream tasks without additional fine-tuning, paving the way for open vocabulary recognition [14]. VLMs have been extended to few-shot classification [66] as well as other tasks beyond image classification such as video understanding [29; 54], image segmentation [30; 52], image generation [43] and 3D reconstruction [63; 7].

### 2.2 Few-shot Adaptation

Few-shot adaptation methods in the context of classification can be categorized into prompt learning and adapter based approaches. Inspired from the recent advances in natural language processing [26], prompt learning aims to learn effective global text or visual prompts for the downstream tasks [66; 65; 23; 2; 61; 58; 47; 6; 45; 49]. Although prompt learning methods have brought significant improvements over the zero-shot baseline, they require back-propagating through the entire text encoder and require having access to the text encoder [38]. On the other hand, adapter-based approaches operate in the feature space and do not require having access to the pre-trained model weights. We distinguish two families of adapter-based approaches. The first one is training-based methods [28; 12; 60] which either train a linear layer [28; 60] or a two-layer MLP such as CLIP-Adapter [12] to perform residual feature blending of the zero-shot classifier.

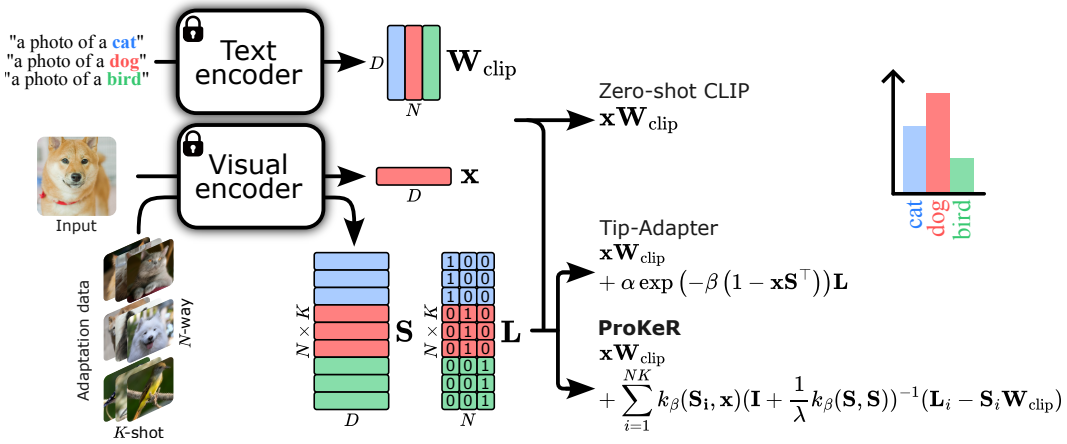


Figure 1: **Overview of our training-free method ProKeR**. While Tip-Adapter builds a key-value cache model using the few-shot samples, ProKeR incorporates a proximal global regularization based on the zero-shot predictor in a reproducing kernel Hilbert space (RKHS). This allows the use of a richer model without overfitting on the few-shot data.

While fine-tuning adapters have achieved a good level of performance, they still require additional training time which is impractical for limited resources and might suffer from the caveats of gradient optimization. To alleviate these issues, Tip-Adapter [62] emerges as a training-free solution based on a key-value cache model. Built on Tip-Adapter, multiple caching methods have been proposed. APE [67] includes a feature selection step to the cache model. CaFo [64] uses an ensemble of foundation models for the cache model and augments text prompts using a Large Language model. SuS-X and its module Tip-X [53] first generates the few-shot set using Stable Diffusion then uses the inter-modal similarities between the test images and the few-shot set. More recently, GDA proposes to use a Linear Discriminant Analysis to set a strong baseline for training-free methods [55]. Despite the good level of performance, there is currently no theoretical framework for better understanding the motivation behind caching models.

## 3 Method

We expose in this section the details of the proposed method. Our starting point consists in framing Tip-Adapter as a kernel method. Thanks to this new perspective, we develop multiple improvements for it. Conclusively, we propose ProKeR, a method that introduces a global proximal regularization in a reproducing kernel Hilbert space (RKHS).

### 3.1 Tip-Adapter as a Nadaraya-Watson estimator

As illustrated in Fig. 1, let $\mathbf{x} \in \mathbb{R}^D$ be the features of an input query image extracted using the visual encoder of CLIP, $\mathbf{S} \in \mathbb{R}^{NK \times D}$ the visual features of the training set and $\mathbf{L} \in \mathbb{R}^{NK \times D}$ the associated matrix of one-hot labels where $N$ is the number of classes and $K$ is the number of shots per class. Let $\mathbf{W}_{\text{clip}} \in \mathbb{R}^{D \times N}$ be the text prototypes of the classes extracted with the text encoder using the standard hand-crafted templates in the form of "a photo of a {CLASS}" [62; 66; 28]. The zero-shot predictor from CLIP is defined as $f_{\text{clip}} : \mathbf{x} \mapsto \mathbf{x}\mathbf{W}_{\text{clip}}$.

To alleviate $f_{\text{clip}}$ prediction errors due to lack of generalization, Tip-Adapter [62] utilizes a cache model to learn knowledge from the few-shot samples. The predicted logits can be formulated as:

$$\phi_{\text{Tip}}(\mathbf{x}) \;=\; f_{\text{clip}}(\mathbf{x}) + \alpha \exp\left(-\beta\left(1 - \mathbf{x}\mathbf{S}^\top\right)\right)\mathbf{L} \tag{1}$$

where $\beta$ is a smoothing scalar and $\alpha$ controls the balance between textual features and few-shot images. Note that since the features in CLIP are normalized (i.e. $||\mathbf{x}||_2 = 1$) , we can rewrite

equation 1 as:

$$\phi_{\text{Tip}}(\mathbf{x}) \;=\; f_{\text{clip}}(\mathbf{x}) + \alpha \sum_{i=1}^{NK} \exp\left(-\frac{\beta}{2}\,||\mathbf{S}_i - \mathbf{x}||_2^2\right)\mathbf{L}_i \tag{2}$$

where $\mathbf{S}_i$ is the $i$-th few-shot sample. Interestingly, the right term of Tip-Adapter can be seen as a modified version of the well-known Nadaraya-Watson (NW) estimator [33] with a Radial Basis Function (RBF) kernel [37]:

$$\phi(\mathbf{x}) \;=\; \frac{\sum_{i=1}^{NK} k_\beta(d(\mathbf{x}, \mathbf{S}_i))\mathbf{L}_i}{\sum_{i=1}^{NK} k_\beta(d(\mathbf{x}, \mathbf{S}_i))} \tag{3}$$

$$\text{where} \quad k_\beta(u) \;=\; \left(\frac{\beta}{2}\right)^D \exp(-\frac{\beta}{2}u) \tag{4}$$

where $d$ is the distance between the query image and the shots in the feature space. In essence, when $d$ is the Euclidean distance, as in equation 2, the adaptation term of Tip-Adapter is a nonparametric regression, obtained by performing a smooth and locally weighted average of the one-hot labels of the few-shot samples using a kernel function, which quantifies the similarity between the query image and the shots.

### 3.2 Training-free few-shot adapters as a Bayes optimal mapping

We formulate the few-shot visual-language adaptation as a Bayes optimal mapping [17] associated to the following pointwise risk.

$$R(\mathbf{x}, \phi(\mathbf{x})) = \mathbb{E}_{Y|X}[s(Y, \phi(X)) + \mathcal{R}_{\text{clip}} \mid X = \mathbf{x}] \tag{5}$$

where $s$ is a cost function and $\mathcal{R}_{\text{clip}}$ is a regularization term using CLIP prediction independent of $Y$. Here, $X$ and $Y$ are random variables representing the image features and the labels respectively.

The pointwise Bayes optimal mapping is defined as [17]:

$$\phi(\mathbf{x}) \;=\; \arg\min_{\mathbf{q}\in M} R(\mathbf{x}, \mathbf{q}) \;=\; \arg\min_{\mathbf{q}\in M} \int_M s(\mathbf{y}, \mathbf{q}) d\mu_{\mathbf{x}}(\mathbf{y}) + \mathcal{R}_{\text{clip}} \tag{6}$$

where $d\mu_{\mathbf{x}}$ is the conditional probability of $Y$ conditioned on $X \;=\; \mathbf{x}$ and $M$ is the output space. Following [17], we leverage kernel estimators to rewrite the adaptation problem as:

$$\phi(\mathbf{x}) \;=\; \arg\min_{\mathbf{q}} \frac{1}{NK} \sum_{i=1}^{NK} k_\beta(d(\mathbf{x}, \mathbf{S}_i)) s(\mathbf{q}, \mathbf{L}_i) + \mathcal{R}_{\text{clip}} \tag{7}$$

This formulation offers a new perspective on the adaptation, where for each test point the cost function is minimized over the output space, with a weighting from each training sample. The regularization term guarantees that the obtained predictions are not far from zero-shot CLIP.

Interestingly, this formulation paves the way to considering different choices of cost functions $s$, regularization terms $\mathcal{R}_{\text{clip}}$ and kernels $k_\beta$. The consistency of the obtained estimator is discussed in [17] where the solution of 7 is obtained using a gradient descent algorithm. While this optimisation can be time consuming, there exists some cases where a closed form solution can be derived. For instance, when $s$ is the squared Euclidean distance and $\mathcal{R}_{\text{clip}} \;=\; \lambda\|\mathbf{q} - f_{\text{clip}}(\mathbf{x})\|_2^2$, we can derive the following solution:

$$\begin{aligned} \phi(\mathbf{x}) \;&=\; \frac{\lambda NK}{\lambda NK + \mathcal{Z}(\mathbf{x})} f_{\text{clip}}(\mathbf{x}) + \frac{1}{\lambda NK + \mathcal{Z}(\mathbf{x})} \sum_{i=1}^{NK} k_\beta(d(\mathbf{x}, \mathbf{S}_i))\mathbf{L}_i \\ \text{where} \quad \mathcal{Z}(\mathbf{x}) &= \sum_{i=1}^{NK} k_\beta(d(\mathbf{x}, \mathbf{S}_i)) \end{aligned} \tag{8}$$

where $\lambda$ is a regularization term that balances between the predictions of zero-shot CLIP and the local fit. The obtained closed form of the estimator in equation 7 is equivalent to Tip-Adapter up to a

scaling factor which depends on each input $\mathbf{x}$. The main difference between the two formulations in equation 2 and 8 is that the second term in equation 8 is agnostic to the training size and is query dependent.

Although the Nadaraya-Waston estimator is a nonparametric model that allows to capture any type of distribution, it is well known to suffer from poor bias at the boundaries of the training samples [36]. In the following, we propose two methods to alleviate this bias.

### 3.3 Local Linear Regression

A standard approach to alleviate the boundary bias of the NW estimator is by moving from a local constant fit to a local linear fit around each test point. Instead of using the estimate from equation 7, local linear regression (LLR) forms the local estimate $\phi(\mathbf{x}) = \tilde{\mathbf{x}}\mathbf{A}$ where $\tilde{\mathbf{x}} = [1 \quad \mathbf{x}]$ and $\mathbf{A} \in \mathbb{R}^{(d+1)c}$ minimizes the following problem:

$$\min_{\mathbf{A}} \frac{1}{NK} \sum_{i=1}^{NK} k_\beta(d(\mathbf{x}, \mathbf{S}_i)) s(\tilde{\mathbf{S}}_i \mathbf{A}, \mathbf{L}_i) + \mathcal{R}_{\text{clip}} \tag{9}$$

which is a weighted ordinary least square problem around each test point weighted by the kernel values $k_\beta(d(\mathbf{x}, \mathbf{S}_i))$. Using the same cost function as for equation 8 and using the regularization term $\mathcal{R}_{\text{clip}} = \lambda \|\tilde{\mathbf{x}}\mathbf{A} - f_{\text{clip}}(\mathbf{x})\|_2^2$, we derive a closed form solution for equation 23 as follows:

$$\phi(\mathbf{x}) = \tilde{\mathbf{x}} \left( \tilde{\mathbf{S}}^\top \mathbf{\Omega} \tilde{\mathbf{S}} + \lambda NK \tilde{\mathbf{x}}^\top \tilde{\mathbf{x}} \right)^{-1} \left( \tilde{\mathbf{S}}^\top \mathbf{\Omega} \mathbf{L} + \lambda NK \tilde{\mathbf{x}}^\top f_{\text{clip}}(\mathbf{x}) \right) \tag{10}$$

where $\mathbf{\Omega}$ is the $NK \times NK$ matrix with $i$th diagonal element as $k_\beta(d(\mathbf{x}, \mathbf{S}_i))$.

This method, that we dub henceforth LLR, boils down to fitting a local linear regression while enforcing the obtained fit to be close to the text predictions locally around the query input sample. While LLR usually improves performance, it is time consuming as one needs to fit a linear regression for each input sample. Furtheremore, the parameters of LLR are solely estimated locally, which is prone to overfitting in high-dimensional problems [36]. Alternatively, one possible way to eliminate the bias of NW estimator due to the bandwidth selection without significantly increasing the time complexity is by equipping the kernel with a better distance function $d$ [36].

### 3.4 Local methods with a global metric

While there exists multiple strategies to construct a good metric [56; 36], using the Mahalanobis distance with the covariance matrix estimated from the training data is a simple yet well-performing one [3]:

$$d(\mathbf{x}, \mathbf{S}_i) = \|\mathbf{x} - \mathbf{S}_i\|_{\hat{\mathbf{\Lambda}}} \tag{11}$$

where $\hat{\mathbf{\Lambda}}$ is the estimated precision matrix from the few-shot samples. This metric effectively incorporates global information and captures the geometry of the space which allows to construct a better kernel function tailored to the downstream task. The classical RBF kernel corresponds to using an isotropic covariance matrix. While still being a local method, adapting the metric to the downstream task allows to incorporate global information about the underlying distribution and improves over Tip-Adapter as shown in Figure 2.

### 3.5 Proximal Kernel Ridge Regression

As can be seen in Figure 2, incorporating global information about the task through a global metric effectively outperforms both Tip-Adapter and LLR. However, the choice of the global metric for NW estimator beyond the Mahalanobis distance remains challenging [56], especially in the few-shot setting. Furthermore, despite being equipped with a global metric, the NW estimator remains in essence a local method and still lacks a global regularization. Whilst the use of a global regularization has been recently addressed in training-based methods [50], using a truly global regularization in a training-free manner remains challenging and unexplored.

These limitations highlight the necessity for regularization in this adaptation process. The main idea is to balance the need to maintain the expressive capacity of the learned functions while ensuring stability and robustness. To this end we devise two important design choices. Firstly, we restrain the hypothesis space of the learned function to be a reproducing kernel Hilbert space (RKHS). Secondly, using the RKHS norm, we introduce a proximal regularization term to ensure that the obtained solution is close to the base predictor i.e. $f_{\text{clip}}$. Thanks to the properties of the RKHS, minimizing the difference between two functions using the RKHS norm ensures that they are close pointwise. Consequently, this proximal term serves as global regularization that preserves prior knowledge from the zero-shot predictor, resulting in more robust solutions that are less prone to overfitting on the few-shot data.

Given the multi-output nature of the problem, the employed reproducing kernel $\mathbf{K}_\beta$ is a reproducing kernel for vector valued functions. The main difference is that the kernel is matrix valued. Several instances of multi-output kernels have been proposed in the literature [5; 1], with separable kernels being among the most widely used for learning vector-valued functions due to their simplicity and computational efficiency. These kernels are formulated as a product of a kernel function for the input space alone and a matrix that encodes the interactions among the outputs. Let us consider $\mathbf{K}_\beta : \mathbb{R}^D \times \mathbb{R}^D \mapsto \mathbb{R}^{N\times N}$ as a separable kernel of the form:

$$(\mathbf{K}_\beta(\mathbf{x}, \mathbf{x}'))_{j,j'} = k_\beta(\mathbf{x}, \mathbf{x}')\mathbf{B} \tag{12}$$

where $\mathbf{B}$ is a $N \times N$ symmetric and positive semi-definite matrix which captures the correlations between the outputs. A simple, yet effective choice for $\mathbf{B}$ is the identity matrix where all outputs are treated as being unrelated.

Our goal is to learn a multi-output predictor $\phi$ using the following objective:

$$\min_{\phi\in\mathcal{H}} \sum_{i=1}^{NK} \|\phi(\mathbf{S_i}) - \mathbf{L}_i\|_2^2 + \lambda\|\phi - f_{\text{clip}}\|_{\mathcal{H}}^2 \tag{13}$$

By the representer theorem [32; 22], the unique minimizer of problem 13 emerges naturally as the solution of a Kernel Ridge Regression (KRR) problem:

$$\begin{aligned} \phi \;&=\; f_{\text{clip}} + \sum_{i=1}^{NK} k_\beta(\mathbf{S_i}, .)\boldsymbol{\gamma}_i \\ \text{where} \quad \boldsymbol{\gamma} \;&=\; (\mathbf{I} + \frac{1}{\lambda} k_\beta(\mathbf{S}, \mathbf{S}))^{-1}(\mathbf{L} - f_{\text{clip}}(\mathbf{S})) \end{aligned} \tag{14}$$

Here, $(k_\beta(\mathbf{S}, \mathbf{S}))_{i,j} = k_\beta(\mathbf{S}_i, \mathbf{S}_j)$ and $\boldsymbol{\gamma}_i \in \mathbb{R}^N$. This approach allows to map data to an infinite dimensional space. Furthermore, the regularization term allows the use of a richer model that captures the complex structure of the data while preserving its smoothness, avoiding overfitting on the few-shot data.

### 3.6 Mercer decomposition of kernel methods

One additional benefit of this perspective on caching methods is memory reduction which allows to overcome the necessity of storing training data. For positive definite kernel, we can leverage Mercer theorem [13] to write the kernel function as follow:

$$k_\beta(\mathbf{x}, \mathbf{x}^{'}) = \psi_\beta(\mathbf{x})\psi_\beta(\mathbf{x}^\top) \tag{15}$$

This allows us to write :

$$\sum_{i=1}^{NK} k_\beta(\mathbf{S_i}, \mathbf{x})\gamma_i = \sum_{i=1}^{NK} \psi_\beta(\mathbf{x})\psi_\beta(\mathbf{S_i})\gamma_i = \psi_\beta(\mathbf{x})[\boldsymbol{\Psi}_\beta^1, \ldots, \boldsymbol{\Psi}_\beta^N] \tag{16}$$

Hence we compute prototypes per class without the need to store additional samples. However, the feature map $\psi$ may not be available in closed form and may be infinite dimensional. For shift invariant kernels like the Gaussian kernel, we leverage the Bochner's theorem following [42] to write:

$$k_\beta(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}') = \mathbb{E}_{\mathbf{w}\sim p(\mathbf{w})}(\psi_\mathbf{w}(\mathbf{x})\psi_\mathbf{w}(\mathbf{x}')^\top) \quad \text{where} \quad \psi(\mathbf{x}) = \exp(i\mathbf{x}\mathbf{w}) \tag{17}$$

where $p(\mathbf{w})$ is the Fourier transform of the kernel $k$. This formulation allows to approximate the RBF kernel with Random Fourier features (RFF). In practice, to lower the variance of the kernel the estimate and thus keep a good balance between performance and the number of RFFs we use Orthogonal Fourier Features [59].

# 4 EXPERIMENTS

In this section, we evaluate our method on multiple image classification benchmarks. We compare our results to existing training-free methods, notably Tip-Adapter [62], which is the baseline of our work. For a fair comparison, we use the same text inputs for all reported methods. We also compare to other state-of-the-art conventional methods, such as APE [67], Tip-X [53], GDA [55], CLIP [41] and CALIP [15] which proposes a parameter-free attention mechanism to improve CLIP in a zero-shot manner. We run APE with the same text templates as Tip-Adapter using their official implementation. Finally, we report the average running times on ImageNet [9] for each method using an NVIDIA RTX A6000 GPU. For completeness, we additionally provide a comparison of our method with existing state-of-the-art training-based methods.

## 4.1 DATASETS AND EVALUATION PROTOCOL

For comprehensive evaluation, we adopt 11 image classification benchmarks: ImageNet [9], Caltech101 [11], DTD [8], EuroSAT [18], FGVCAircraft [31], Flowers102 [35], Food101 [4], OxfordPets [40], StanfordCars [24], SUN397 [57], and UCF101 [51]. For testing the generalization ability of our method, we further test on ImageNet-Sketch [48] and ImageNet-V2 [44].

For a fair comparison with previous works, we use ResNet-50 for the visual encoder of CLIP unless mentioned otherwise. We follow two settings for our experiments. The first setting, initially introduced by [50] for training-based methods, consists of selecting the best hyperparameters of each method on ImageNet and transfer them to the other datasets and report the average performance. This setting reflects real-life scenarios where a validation set may not be available especially in a few-shot regime. The second setting follows CoOp's benchmark [66] where validation shots are used to select the hyperparameters and evaluate the results on the full test set.

## 4.2 EXPERIMENT RESULTS AND ANALYSIS

### 4.2.1 COMPARISON WITH ALTERNATIVES TO TIP-ADAPTER

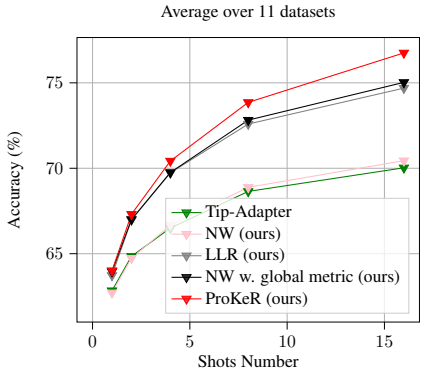


Figure 2: **Average performance for different methods on 11 image classification datasets.**

We compare in Figure 2 our result when improving Tip-Adapter using kernel based approaches across 11 datasets. Our reformulation in equation 8 outperforms Tip-Adapter for 8 and 16 shots and maintains the same level of performance in the lower shot setting. We argue that this is due to the fact that our reformulation is agnostic to the training size. Debiasing the NW estimator using our regularized LLR in equation 23 significantly improves the performance as a first order polynomial fit compared to the constant fit of the NW estimator. Additionally, the use of the Mahalanobis distance

as global metric for NW estimator further increases the performance especially with more shots as the estimation of the precision matrix from the training set becomes more accurate. Furthermore, by introducing a proximal regularization in the RKHS, our method ProKeR significantly outperforms Tip-Adapter as well as all the proposed alternatives especially with more shots.

#### 4.2.2 COMPARISON WITH STATE-OF-THE-ART

We report in Table 1 the performance of different methods in a realistic and practical validation-free experimental setting. First, our method ProKeR outperforms on average existing alternatives by a far margin (2.04% compared to the second best), especially in the low shot regimes (1, 2 and 4) and is only outperformed in the 8 shot setting. Furthermore, when using a Polynomial kernel (defined in Table 4), our method sets a new standard in the 16 shots regime.

Table 1: Average performance on 11 classification datasets for different shots. Hyperparameters are transfered from ImageNet.

| Shots | 1 | 2 | 4 | 8 | 16 | Average |
|---|---|---|---|---|---|---|
| Tip-Adapter [62] | 58.86 | 60.33 | 61.49 | 63.15 | 64.61 | 61.68 |
| APE [67] | 60.09 | 62.33 | 65.36 | 67.95 | 69.89 | 65.12 |
| GDA [55] | 57.49 | 63.43 | 66.68 | **72.46** | 75.12 | 67.03 |
| ProKeR (Polynomial) (ours) | 62.39 | 65.59 | 68.05 | 71.81 | **75.82** | 68.72 |
| ProKeR (RBF) (ours) | **63.13** | **66.31** | **68.64** | 72.15 | 75.12 | **69.07** |

In the CoOp's benchmark where validation shots are used to tailor hyperparamters for each dataset, our method ProKer surpasses existing training-free methods as shown in Table 2. These superior results fully validate the significance of using the global regularization in the RKHS.

Table 2: Performance on 11 different classification datasets (CoOp's benchmark).

| Shots | 1 | 2 | 4 | 8 | 16 | Average |
|---|---|---|---|---|---|---|
| GDA [55] | 62.19 | 66.19 | 69.77 | 73.30 | 76.04 | 69.49 |
| Tip-Adapter [62] | 62.83 | 64.84 | 66.50 | 68.64 | 70.02 | 66.56 |
| Tip-Adapter [62] with RFF | 62.77 | 64.58 | 66.29 | 68.15 | 69.60 | 66.27 |
| APE [67] | **64.43** | 66.48 | 68.66 | 70.75 | 72.81 | 68.62 |
| APE [67] with RFF | 64.16 | 66.11 | 68.76 | 70.37 | 72.02 | 68.28 |
| ProKeR (ours) | 64.01 | **67.31** | **70.42** | **73.85** | **76.75** | **70.46** |
| ProKeR with RFF (ours) | 64.01 | 67.12 | 70.28 | 73.61 | 76.44 | 70.29 |

#### 4.2.3 GENERALIZATION ABILITY

Table 3: **Robustness to distribution shift of different methods for 16 shots.**

| Datasets | **Source** | **Target** | | | | |
|---|---|---|---|---|---|---|
| | ImageNet [9] | -V2 [44] | -Sketch [48] | -A [20] | -R [19] | Average |
| Zero-Shot CLIP [41] | 60.33 | 53.27 | 35.44 | 23.61 | 60.42 | 46.61 |
| Tip-Adapter [62] | 61.43 | 54.13 | 35.71 | **23.63** | 60.41 | 47.06 |
| APE [67] | 62.60 | 54.93 | 35.41 | 22.95 | 59.90 | 47.15 |
| GDA [55] | 63.82 | 55.35 | 34.32 | 19.53 | 55.56 | 45.71 |
| ProKeR (Polynomial) (Ours) | **64.66** | **56.11** | **36.08** | 23.27 | 60.55 | **48.13** |
| ProKeR (Ours) | 64.45 | 56.02 | **36.08** | 23.37 | **60.59** | 48.10 |

In Table 3, we test the generalization ability of the different methods on out-of-distribution datasets. Notably, the shots are drawn from ImageNet and the test set is drawn from either ImageNet-V2 or

ImageNet-Sketch. ProKeR achieves state-of-the-art performance for training-free methods on both in-distribution and out-of-distribution datasets.

### 4.3 KERNEL ABLTATION

So far, we have performed our analysis using the RBF kernel, a commonly used kernel in the kernel literature and in cache-based methods. Nevertheless, through the lens of our kernel perspective on cache-based, different kernels can be considered ranging from a linear kernels to more elaborate ones. We perform in Table 4 an ablation study where we discuss different kernel choices. Besides the RBF kernel, we consider three commonly used kernels: the Linear kernel, the Epanechnikov kernel and the Polynomial kernel. The RBF kernel outperforms the linear, Epanechnikov and the polynomial kernels. Using the RBF kernel allows us to project data into an infinite dimensional space which captures more complex relationships.

| Kernel | $k(\mathbf{x}, \mathbf{x}')$ | Accuracy |
|---|---|---|
| Linear | $\mathbf{x}\mathbf{x}'^{\top}$ | 72.34 |
| Epanechnikov | $\frac{3}{4}\left(1 - \|\mathbf{x} - \mathbf{y}'\|_2^2\right)$ | 74.43 |
| Polynomial | $\left(\mathbf{x}\mathbf{y}'^{\top}\right)^2$ | 76.61 |
| RBF | $\exp(-\frac{\beta}{2}\|(\mathbf{x} - \mathbf{y}'\|_2^2))$ | **76.75** |

Table 4: Kernel Ablation for 16 shots on 11 datasets on CoOp's benchmark.

### 4.4 ABLATION ON CLIP ARCHITECTURES

In Table 5, we report the performance of training-free methods using different backbones on ImageNet for 16 shots. Our method consistently performs better than the alternatives across all architectures. While all methods improve with a more capable architectures, the gap between our method and the second best one (APE) remains stable.

Table 5: **Average performance on 16-shot ImageNet with different backbones.**

| Models | ResNet-50 | ResNet-101 | ViT-B/32 | ViT-B/16 |
|---|---|---|---|---|
| Zero-shot CLIP [41] | 60.33 | 62.53 | 63.80 | 68.73 |
| Tip-Adapter [62] | 61.43 | 64.08 | 65.18 | 70.25 |
| APE [67] | 62.60 | 65.61 | 66.31 | 71.37 |
| ProKeR (Ours) | **64.45** | **67.39** | **68.12** | **73.25** |

### 4.5 ADRESSING MEMORY LIMITATIONS OF CACHE-BASED METHODS

In Table 2, we report the performance when using Random Fourier features to alleviate the memory limitations of cache-based methods. Using RFFs, we're able to drastically reduce the memory print of caching methods while maintaining almost the same performance across different shots. Our method when combined with RFFs still maintains state-of-the-art performance.

### 4.6 RUNNING TIMES AND MEMORY REQUIREMENTS

Next, we report running time (train and test) for different methods as well as the memory requirements for each method. Our method is on part with Tip-Adapter and GDA in term of speed. Note that APE runs a feature selection step which takes additional time to run, the third position. On the other hand, training-based methods are orders of magnitude slower. Regarding the memory complexity, ProKeR stores the training shots similarly to APE and Tip-Adapter. However, when using Random Fourier features, our method does need to store additional training samples.

Table 6: **Running times on ImageNet for 16 shots.** All experiments are performed on a RTX A6000 GPU. For each method we report the memory requirements. $D_1$ is the inner dimension of the MLP used in Clip-Adapter [12] and $T$ is the number of text tokens in CoOp [66]. $R$ is the number of Fourier features used tp approximate the RBF kernel.

| Methods | Overall Time | Train | Memory requirements |
|---|---|---|---|
| CoOp[66] | $\sim$ 17h | ✓ | $N \times T \times D$ |
| Clip-Adapter[12] | $\sim$ 40min | ✓ | $(2N + D_1) \times D$ |
| CrossModal-LP[28] | $\sim$ 3min | ✓ | $N \times D$ |
| Standard LP[41] | $\sim$ 3min | ✓ | $N \times D$ |
| Tip-adapter-F[62] | $\sim$ 7min | ✓ | $N \times K + N \times (K + 1) \times D$ |
| Tip-Adapter[62] | 2.1s | ✗ | $N \times K + N \times (K + 1) \times D$ |
| APE [67] | 24.6s | ✗ | $N \times K + N \times (K + 1) \times D$ |
| GDA [55] | 1.6 s | ✗ | $2 \times N \times D$ |
| ProKeR (Ours) | 4.7s | ✗ | $N \times K + N \times (K + 1) \times D$ |
| ProKeR with RFF (Ours) | 4.2s | ✗ | $N \times (D + R)$ |

## 5 Limitations & Future Work

Based on our analysis, caching methods can be understood as local nonparametric regressors. These methods lack a global regularization from zero-shot CLIP which limits their generalization ability when performing adaptation. On the other hand, global methods may lack the flexibility when dealing with complex data. In the future work, we will explore how both local and global methods can be combined to benefit from the best of both worlds. Furthermore, our formulation of caching methods as a Nadaraya-Watson estimator offers multiple options for the choice the regularization term, the metric used in the kernel as well as the bandwidth selection [39]. These choices constitute different ways to reduce the bias-variance trade-off inherent to these methods [16].

## 6 Conclusion

In this paper, we propose a theoretical understanding of Tip-Adapter, a training-free caching-based method. Our analysis suggests that Tip-Adapter is a local nonparametric regression that has well-known bias limitation. We propose multiple angles of improvement that has shown significant amelioration over Tip-Adapter's baseline. Subsequently, we demonstrate that incorporating global information in a training-free method can be achieved using a global regularization in a reproducing kernel Hilbert space (RKHS), which conclusively further improves the state-of-the-art for training-free methods.

## REFERENCES


[1] Mauricio A Alvarez, Lorenzo Rosasco, Neil D Lawrence, et al. Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning*, 4(3):195–266, 2012. 6

[2] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Visual prompting: Modifying pixel space to adapt pre-trained models. *arXiv preprint arXiv:2203.17274*, 2022. 2

[3] Peyman Bateni, Raghav Goyal, Vaden Masrani, Frank Wood, and Leonid Sigal. Improved few-shot visual classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14493–14502, 2020. 5

[4] 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*, pp. 446–461. Springer, 2014. 7

[5] Andrea Caponnetto, Charles A Micchelli, Massimiliano Pontil, and Yiming Ying. Universal multi-task kernels. *The Journal of Machine Learning Research*, 9:1615–1646, 2008. 6

[6] Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. Plot: Prompt learning with optimal transport for vision-language models. *arXiv preprint arXiv:2210.01253*, 2022. 2

[7] Runnan Chen, Youquan Liu, Lingdong Kong, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, Yu Qiao, and Wenping Wang. Clip2scene: Towards label-efficient 3d scene understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7020–7030, 2023. 2

[8] 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*, pp. 3606–3613, 2014. 7

[9] 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*, pp. 248–255. Ieee, 2009. 7, 8

[10] Alex Fang, Gabriel Ilharco, Mitchell Wortsman, Yuhao Wan, Vaishaal Shankar, Achal Dave, and Ludwig Schmidt. Data determines distributional robustness in contrastive language image pre-training (clip). In *International Conference on Machine Learning*, pp. 6216–6234. PMLR, 2022. 1

[11] 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*, pp. 178–178. IEEE, 2004. 7

[12] 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. *arXiv preprint arXiv:2110.04544*, 2021. 1, 2, 10

[13] Benyamin Ghojogh, Ali Ghodsi, Fakhri Karray, and Mark Crowley. Reproducing kernel hilbert space, mercer's theorem, eigenfunctions, nystr\" om method, and use of kernels in machine learning: Tutorial and survey. *arXiv preprint arXiv:2106.08443*, 2021. 6

[14] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021. 2

[15] Ziyu Guo, Renrui Zhang, Longtian Qiu, Xianzheng Ma, Xupeng Miao, Xuming He, and Bin Cui. Calip: Zero-shot enhancement of clip with parameter-free attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 746–754, 2023. 7

[16] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009. 2, 10

[17] Matthias Hein. Robust nonparametric regression with metric-space valued output. *Advances in neural information processing systems*, 22, 2009. 4

[18] 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. 7

[19] 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*, pp. 8340–8349, 2021. 8

[20] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021. 8

[21] 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*, pp. 4904–4916. PMLR, 2021. 1, 2

[22] Hachem Kadri, Emmanuel Duflos, Philippe Preux, Stéphane Canu, Alain Rakotomamonjy, and Julien Audiffren. Operator-valued kernels for learning from functional response data. *Journal of Machine Learning Research*, 17(20):1–54, 2016. 6

[23] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[24] 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*, pp. 554–561, 2013. 7

[25] Tam Le, Truyen Nguyen, Makoto Yamada, Jose Blanchet, and Viet Anh Nguyen. Adversarial regression with doubly non-negative weighting matrices. *Advances in Neural Information Processing Systems*, 34:16964–16976, 2021. 1

[26] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 2

[27] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *arXiv preprint arXiv:2110.05208*, 2021. 2

[28] Zhiqiu Lin, Samuel Yu, Zhiyi Kuang, Deepak Pathak, and Deva Ramanan. Multimodality helps unimodality: Cross-modal few-shot learning with multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19325–19337, 2023. 1, 2, 3, 10, 16

[29] Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard De Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen clip models are efficient video learners. In *European Conference on Computer Vision*, pp. 388–404. Springer, 2022. 1, 2

[30] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7086–7096, 2022. 1, 2

[31] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 7

[32] Charles A Micchelli and Massimiliano Pontil. On learning vector-valued functions. *Neural computation*, 17(1):177–204, 2005. 6

[33] Elizbar A Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1): 141–142, 1964. 1, 4

[34] Thao Nguyen, Gabriel Ilharco, Mitchell Wortsman, Sewoong Oh, and Ludwig Schmidt. Quality not quantity: On the interaction between dataset design and robustness of clip. *Advances in Neural Information Processing Systems*, 35:21455–21469, 2022. 1

[35] 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*, pp. 722–729. IEEE, 2008. 7

[36] Yung-Kyun Noh, Masashi Sugiyama, Kee-Eung Kim, Frank Park, and Daniel D Lee. Generative local metric learning for kernel regression. *Advances in neural information processing systems*, 30, 2017. 1, 2, 5

[37] Mark JL Orr et al. Introduction to radial basis function networks, 1996. 4

[38] Yassine Ouali, Adrian Bulat, Brais Matinez, and Georgios Tzimiropoulos. Black box few-shot adaptation for vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15534–15546, 2023. 2

[39] Byeong Park and Berwin Turlach. Practical performance of several data driven bandwidth selectors. Technical report, Université catholique de Louvain, Center for Operations Research and . . . , 1992. 10

[40] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012. 7

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html. 1, 2, 7, 8, 9, 10

[42] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007. 6

[43] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pp. 8821–8831. Pmlr, 2021. 2

[44] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pp. 5389–5400. PMLR, 2019. 7, 8

[45] Shuvendu Roy and Ali Etemad. Consistency-guided prompt learning for vision-language models. *arXiv preprint arXiv:2306.01195*, 2023. 2

[46] David Ruppert and Matthew P Wand. Multivariate locally weighted least squares regression. *The annals of statistics*, pp. 1346–1370, 1994. 2

[47] Jameel Hassan Abdul Samadh, Hanan Gani, Noor Hazim Hussein, Muhammad Uzair Khattak, Muzammal Naseer, Fahad Khan, and Salman Khan. Align your prompts: Test-time prompting with distribution alignment for zero-shot generalization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2

[48] Hengcan Shi, Munawar Hayat, Yicheng Wu, and Jianfei Cai. Proposalclip: Unsupervised open-category object proposal generation via exploiting clip cues. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9611–9620, 2022. 7, 8

[49] 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. 2

[50] Julio Silva-Rodriguez, Sina Hajimiri, Ismail Ben Ayed, and Jose Dolz. A closer look at the few-shot adaptation of large vision-language models. *arXiv preprint arXiv:2312.12730*, 2023. 2, 5, 7, 16

[51] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 7

[52] Zeyi Sun, Ye Fang, Tong Wu, Pan Zhang, Yuhang Zang, Shu Kong, Yuanjun Xiong, Dahua Lin, and Jiaqi Wang. Alpha-clip: A clip model focusing on wherever you want. *arXiv preprint arXiv:2312.03818*, 2023. 2

[53] Vishaal Udandarao, Ankush Gupta, and Samuel Albanie. Sus-x: Training-free name-only transfer of vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2725–2736, 2023. 1, 3, 7

[54] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *arXiv preprint arXiv:2109.08472*, 2021. 2

[55] 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. 3, 7, 8, 10

[56] Kilian Q Weinberger and Gerald Tesauro. Metric learning for kernel regression. In *Artificial intelligence and statistics*, pp. 612–619. PMLR, 2007. 5

[57] 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*, pp. 3485–3492. IEEE, 2010. 7

[58] Hantao Yao, Rui Zhang, and Changsheng Xu. Visual-language prompt tuning with knowledge-guided context optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6757–6767, 2023. 2

[59] Felix Xinnan X Yu, Ananda Theertha Suresh, Krzysztof M Choromanski, Daniel N Holtmann-Rice, and Sanjiv Kumar. Orthogonal random features. *Advances in neural information processing systems*, 29, 2016. 7

[60] Tao Yu, Zhihe Lu, Xin Jin, Zhibo Chen, and Xinchao Wang. Task residual for tuning vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10899–10909, 2023. 2, 16

[61] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Unified vision and language prompt learning. *arXiv preprint arXiv:2210.07225*, 2022. 2

[62] Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021. 1, 2, 3, 7, 8, 9, 10, 16

[63] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8552–8562, 2022. 2

[64] Renrui Zhang, Xiangfei Hu, Bohao Li, Siyuan Huang, Hanqiu Deng, Yu Qiao, Peng Gao, and Hongsheng Li. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15211–15222, 2023. 3

[65] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2

[66] 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. 1, 2, 3, 7, 10

[67] Xiangyang Zhu, Renrui Zhang, Bowei He, Aojun Zhou, Dong Wang, Bin Zhao, and Peng Gao. Not all features matter: Enhancing few-shot clip with adaptive prior refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2605–2615, 2023. 1, 3, 7, 8, 9, 10, 16

# A SUPPLEMENTAL MATERIAL

## A.1 DETAILED DERIVATIONS

### A.1.1 NADARAYA-WATSON ESTIMATOR

We first derive the solution of the adaptation problem for the Nadaraya-Waston estimator in equation 8. The adaptation problem writes as:

$$\phi(\mathbf{x}) \;=\; \arg\min_{\mathbf{q}} \frac{1}{NK}\sum_{i=1}^{NK} k_\beta(d(\mathbf{x},\mathbf{S}_i))\|\mathbf{q}-\mathbf{L}_i\|_2^2 + \|\mathbf{q}-f_{\text{clip}}(\mathbf{x})\|_2^2 \tag{18}$$

The derivation of the solution of equation 18 is as follows:

$$\text{Let} \quad \mathcal{L} = \frac{1}{NK}\sum_{i=1}^{NK} k_\beta(d(\mathbf{x},\mathbf{S}_i))\|\mathbf{q}-\mathbf{L}_i\|_2^2 + \|\mathbf{q}-f_{\text{clip}}(\mathbf{x})\|_2^2 \tag{19}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{q}} = 0 \Rightarrow \frac{1}{NK}\sum_{i=1}^{NK} k_\beta(d(\mathbf{x},\mathbf{S}_i))\,(\mathbf{q}-\mathbf{L}_i) + \lambda\mathbf{q} - \lambda f_{\text{clip}}(\mathbf{x}) = 0 \tag{20}$$

$$\Rightarrow \mathbf{q}\left(\lambda NK + \sum_{i=1}^{NK} k_\beta(d(\mathbf{x},\mathbf{S}_i))\right) = \lambda NK f_{\text{clip}}(\mathbf{x}) + \sum_{i=1}^{NK} k_\beta(d(\mathbf{x},\mathbf{S}_i))\mathbf{L}_i \tag{21}$$

$$\Rightarrow \mathbf{q} \;=\; \frac{\lambda NK}{\lambda NK + \sum_{i=1}^{NK} k_\beta(d(\mathbf{x},\mathbf{S}_i))} f_{\text{clip}}(\mathbf{x}) + \frac{1}{\lambda NK + \sum_{i=1}^{NK} k_\beta(d(\mathbf{x},\mathbf{S}_i))} \sum_{i=1}^{NK} k_\beta(d(\mathbf{x},\mathbf{S}_i))\mathbf{L}_i \tag{22}$$

### A.1.2 LOCAL LINEAR REGRESSION

Here, we detail the derivation of the solution of the local linear regression (LLR) in equation 10. Let $\tilde{\mathbf{x}} \;=\; [1 \quad \mathbf{x}]$ and $\mathbf{A} \in \mathbb{R}^{(d+1)c}$ which minimizes the following problem:

$$\min_{\mathbf{A}} \frac{1}{NK}\sum_{i=1}^{NK} k_\beta(d(\mathbf{x},\mathbf{S}_i))\|\tilde{\mathbf{S}}_i\mathbf{A}-\mathbf{L}_i\|_2^2 + \lambda\|\tilde{\mathbf{x}}\mathbf{A} - f_{\text{clip}}(\mathbf{x})\|_2^2 \tag{23}$$

Let $\mathbf{\Omega}$ be the $NK \times NK$ matrix with $i$th diagonal element as $k_\beta(d(\mathbf{x},\mathbf{S}_i))$. The derivation is as follows:

$$\text{Let} \quad \mathcal{L} = \frac{1}{NK}\mathbf{\Omega}\|\tilde{\mathbf{S}}\mathbf{A}-\mathbf{L}\|_2^2 + \lambda\|\tilde{\mathbf{x}}\mathbf{A} - f_{\text{clip}}(\mathbf{x})\|_2^2 \tag{24}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{A}} = 0 \Rightarrow \frac{1}{NK}\tilde{\mathbf{S}}^\top\mathbf{\Omega}\left(\tilde{\mathbf{S}}\mathbf{A}-\mathbf{L}\right) + \lambda\tilde{\mathbf{x}}^\top\left(\tilde{\mathbf{x}}\mathbf{A} - f_{\text{clip}}(\mathbf{x})\right) = 0 \tag{25}$$

$$\Rightarrow \left(\tilde{\mathbf{S}}\mathbf{\Omega}\tilde{\mathbf{S}} + \lambda NK\tilde{\mathbf{x}}^\top\tilde{\mathbf{x}}\right)\mathbf{A} = \tilde{\mathbf{S}}^\top\mathbf{\Omega}\mathbf{L} + \lambda NK\tilde{\mathbf{x}}^\top f_{\text{clip}}(\mathbf{x}) \tag{26}$$

$$\Rightarrow \mathbf{A} = \left(\tilde{\mathbf{S}}\mathbf{\Omega}\tilde{\mathbf{S}} + \lambda NK\tilde{\mathbf{x}}^\top\tilde{\mathbf{x}}\right)^{-1}\left(\tilde{\mathbf{S}}^\top\mathbf{\Omega}\mathbf{L} + \lambda NK\tilde{\mathbf{x}}^\top f_{\text{clip}}(\mathbf{x})\right) \tag{27}$$

### A.2 Sensitivity Analysis of $\lambda$

We analyze the sensitivity of $\lambda$ in Table 7. We compute the average value for each dataset and study the effect of varying its value. Overall, the value of lambda is quite stable in a range of 1/3 of its value up to 3 times its value with only a drop of 1.2% accuracy. Varying lambda up to a fifth or 5 times its value only leads to a drop of 3%.

Table 7: Sensitivity Analysis of $\lambda$ on 11 datasets for 16-shots.

| | $\lambda \times 5$ | $\lambda \times 4$ | $\lambda \times 3$ | $\lambda \times 2$ | $\lambda$ | $\lambda \times 2$ | $\lambda \times 3$ | $\lambda \times 4$ | $\lambda \times 5$ | ProKeR |
|---|---|---|---|---|---|---|---|---|---|---|
| Average | 73.17 | 74.23 | 75.24 | 76.27 | 76.58 | 75.85 | 75.11 | 74.45 | 73.84 | 76.75 |

### A.3 Comparison per dataset

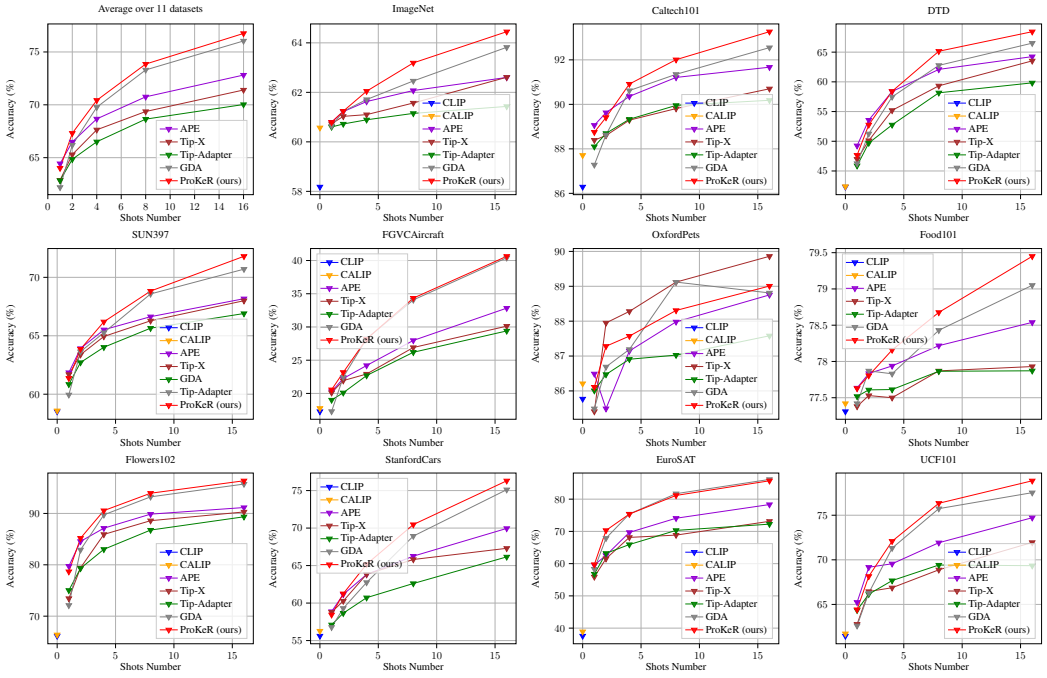


Figure 3: **Few-shot Performance of Training-free Methods** on 11 image classification datasets (CoOp's benchmark).

### A.4 Comparison with training-based methods

Table 8: Performance on 11 classification datasets for 16 shots. Hyperparameters are transfered from ImageNet.

| Method | ImageNet | Caltech101 | OxfordPets | StanfordCars | Flowers102 | Food101 | FGVC | SUN397 | DTD | EuroSAT | UCF101 | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Tip-Adapter-F [62] | 62.27 | 91.22 | 85.43 | 69.56 | 91.18 | 74.65 | 29.32 | 68.90 | 64.56 | 76.55 | 71.81 | 71.40 |
| CrossModalLP [28] | 52.90 | 92.77 | 87.48 | 75.44 | 95.20 | 77.14 | 33.30 | 70.56 | 66.92 | 82.03 | 76.40 | 73.65 |
| TaskRes [60] | 60.85 | 93.09 | 86.28 | 75.38 | 96.14 | 75.43 | 36.53 | 68.43 | 65.88 | 83.70 | 76.96 | 74.42 |
| APE-T [67] | 63.06 | 91.83 | 87.93 | 70.32 | 93.93 | 77.65 | 34.17 | 64.47 | 64.77 | 81.94 | 76.11 | 68.28 |
| CLAP [50] | **65.02** | 91.93 | **88.51** | 75.12 | **94.21** | 78.55 | 33.59 | 70.78 | 66.41 | 80.07 | 76.29 | 74.57 |
| ProKeR (Polynomial) (ours) | 64.66 | **93.36** | 88.14 | **75.68** | 92.84 | **79.26** | **37.37** | **71.47** | 67.43 | **85.37** | **78.50** | **75.82** |
| ProKeR (ours) | 63.77 | 93.23 | 88.15 | 74.58 | 90.62 | 79.14 | 35.33 | 71.44 | **67.49** | 84.49 | 78.00 | 75.11 |

We report in Table 8 the comparison of our method with training-based methods. While being training-free, our method ProKeR outperforms both existing training-based methods on 8 out of 11 datasets and outperforms the second best on average by 1.25%. This emphasizes the effectiveness of incorporating a global regularization using the zero-shot predictor in a reproducing kernel Hilbert space (RKHS).