OT-VP: Optimal Transport-guided Visual Prompting for Test-Time Adaptation

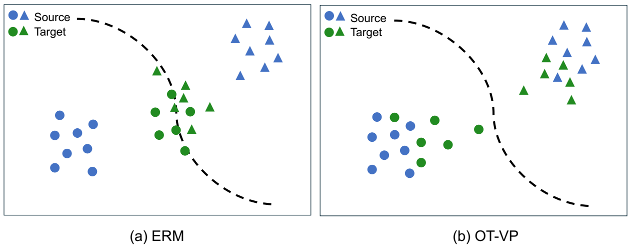
Yunbei Zhang[1], Akshay Mehra[1], Jihun Hamm[1]
[1]Tulane University
{yzhang111, amehra, jhamm3}@tulane.edu

Abstract

While Vision Transformers (ViTs) have demonstrated remarkable capabilities in learning representations, their performance is compromised when applied to unseen domains. Previous methods either engage in prompt learning during the training phase or modify model parameters at test time through entropy minimization. The former often overlooks unlabeled target data, while the latter doesn't fully address domain shifts. In this work, our approach, Optimal Transport-guided Test-Time Visual Prompting (OT-VP), handles these problems by leveraging prompt learning at test time to align the target and source domains without accessing the training process or altering pre-trained model parameters. This method involves learning a universal visual prompt for the target domain by optimizing the Optimal Transport distance. With just four prompt tokens learned, OT-VP achieves a 5.0% *and* 1.5% *increase in averaged accuracy across single-source and multi-source settings on three benchmark datasets, which is* $1.2\times$ *and* $1.5\times$ *the improvement of the state-of-the-art method, respectively.*

1. Introduction

The remarkable successes of Deep Neural Networks (DNNs) are often tempered by the challenges posed by discrepancies between training and testing data distributions [14, 19, 39]. Such discrepancies are not uncommon in real-world applications, where variations in data due to natural differences and stylistic changes can significantly impact model performance [21]. Though Domain Generalization (DG) has been proposed as a solution to ensure model robustness across unseen domains [4, 50], fully achieving this remains a challenge. To address this limitation, a new research direction has emerged, concentrating on enhancing model performance directly at test time [15, 47]. This approach allows models to leverage unlabeled test data from target domains. This data offers insights into the target distribution, insights that are typically inaccessible in the DG framework. Test-time adaptation, as demonstrated in [15], surpasses the capabilities of many existing DG strategies by utilizing immediate, real-world data to refine model accuracy. Inspired by these insights, our work pivots towards exploring test-time adaptation (TTA) as a strategic response to the challenges of domain shifts, aiming to harness the full potential of DNNs in unseen environments.



(a) ERM

(b) OT-VP

Figure 1. Motivation of our approach. (a) An ERM model trained on the source domain struggles to adapt to the target domain due to domain shifts. (b) Our method (OT-VP) optimizes a visual prompt by minimizing the Optimal Transport distance to align the target domain with the source domain without changing the decision boundary.

Our approach is underpinned by leveraging visual prompt learning, designed to seamlessly bridge the gap between source and target domains during test time. Vision Transformers (ViTs), known for their remarkable achievements across a spectrum of computer vision tasks, serve as the backbone of our approach [7, 25]. The self-attention mechanism within ViTs enables comprehensive modeling of relationships between various segments of an image, making them an ideal foundation for our work [33, 38, 49]. Visual prompt learning emerges as a prominent strategy for fine-tuning ViTs without altering model parameters on specialized downstream tasks [16]. This technique involves embedding task-specific knowledge directly into the input tokens, allowing for interaction with prompt tokens via self-attention layers. Such interactions enable the network to grasp the essence of the task, provided the prompt contains ample informative content. Although visual prompt tuning is lauded for its precision in task-oriented learning, it conventionally depends on the availability of labeled data for the creation of impactful prompt tokens—a requirement not met in the TTA context.

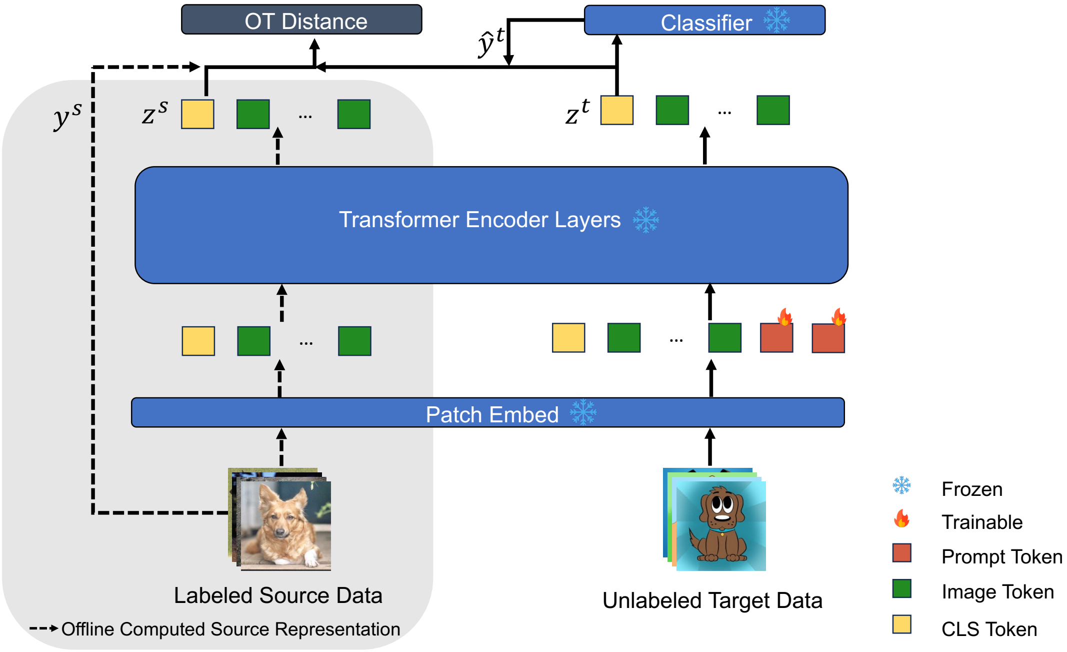


Figure 2. An overview of our proposed OT-VP method. At test time, unlabeled target data are processed through a frozen pre-trained ViT model with only the prompt tokens (indicated in red) being trainable. This generates target representations (z^t) and pseudo-labels ($\hat{y}^t$). We then align these with actual source labels (y^s) and offline-computed (the grey shadowed area) source representations (z^s) via Optimal Transport (OT) distance. The visual prompts are iteratively optimized based on this distance to align the source and target domain data more closely.

Current approaches in prompt learning typically incorporate prompts during the training phase, leveraging source data to create representative prompts [49], or tuning prompts based on source domains at test time [10] for application in target domains. However, these conventional strategies often do not directly address the distribution shifts observed in the target domain, shown in Fig. 1 (a). Meanwhile, recent efforts focusing on prompt learning at test time [27, 40], are specifically designed for the CLIP (Contrastive Language-Image Pretraining) [37] model, targeting Vision-Language models exclusively. These methods generate input-specific prompts based on a limited set of augmented target data, potentially missing the broader differences between source and target domains. Adopting a universal prompt for the target domain leverages more data and reduces computation time. Yet, simply learning this universal prompt by minimizing entropy doesn't effectively tackle domain shifts or enhance performance, as evidenced by our experiments in Sec. 4.3.

To enable ViT models to adapt effectively at test time, it's imperative to address the fundamental challenge of reducing the distribution gap between source and target domains. As shown in Fig. 1, this step is crucial for enhancing the model's ability to generalize to new, unseen data. Many metrics, such as Maximum Mean Discrepancy (MMD), Kullback-Leibler (KL) divergence, and Optimal Transport (OT) distance, have been explored to address distribution shifts [3, 12, 20, 28]. Among these, Optimal Transport (OT) is distinguished as particularly effective due to its ability to utilize the geometry of the underlying space [35]. Numerous studies have demonstrated a strong correlation between OT distance and target performance, supported by thorough theoretical analyses [29, 42, 45]. Inspired by these insights, we propose Optimal Transport [17, 35] guided Test-Time Visual Prompting (OT-VP), a test-time distribution alignment method using prompt learning. Specifically, OT-VP explicitly aligns the Optimal Transport distance between the source representation, computed offline, and the target representation without accessing the training process and modifying the pre-trained model parameters. As illustrated in Fig. 2, for a given target dataset, we pass the unlabeled target image through the visual encoder with learnable prompts to get the target representation and pseudo-labels. Then we update the prompts by minimizing the OT distance. This process is repeated iteratively to ensure that the prompts are well-aligned with the target distribution.

We comprehensively evaluate the efficacy of OT-VP across three standard benchmarks. Beyond the conventional multi-source DG setting—which leaves one domain out as the target while treating the remainder as sources—we also explore a single-source setting. The latter setting, often overlooked in existing literature [15, 49], involves using one domain as the source and a different one as the target. Training on a smaller set makes adaptation to the new target domain more challenging, providing a rigorous test of the model's generalization ability. Based on ViT architecture, OT-VP achieves a stable and significant improvement in ERM, enhancing average accuracy across all three datasets by 5.0% and 1.5% in single-source and multi-source settings respectively. Notably, OT-VP surpasses state-of-the-art (SOTA) algorithms in both single-source and multi-source settings, underscoring its effectiveness and adaptability. Additionally, our method effectively reduces prediction entropy without directly minimizing it, thereby boosting model confidence and reliability on unseen data. Our contribution can be summarized as follows:

- We propose OT-VP, a novel test-time adaptation that reduces the domain gap between source and target through minimizing Optimal Transport distance, facilitating prompts to better adapt to unlabelled target tasks.
- To the best of our knowledge, we are the first to learn universal prompts in the TTA setting. Our findings demonstrate that a universal prompt for a target domain can effectively align it with the source domain and enhance performance.
- We demonstrate through extensive experimentation that OT-VP consistently improves the performance of pre-trained models across a variety of settings, outperforming existing SOTA methods.

2. Related Work

Prompting for Vision Transformers. Vision Transformers (ViTs) have achieved state-of-the-art results in image classification [7, 25], yet adapting ViTs to unseen target domains without labels is still a challenge. Many recent methods enhance ViTs' transferability by learning visual prompts as continuous, learnable vectors trained in an end-to-end manner with frozen model parameters [2, 16, 48]. However, these methods typically require labeled target data for prompt learning. For example, [49] introduced DoPrompt, a DG method that learns visual prompts for source domains and generates input-specific prompts at test time. However, its practical use is limited because it requires modifications to the training process. In contrast, our approach learns a universal prompt for the target domain, effectively minimizing the distribution shift between source and target domains without accessing the training setup.

Test-Time Adaptation (TTA). TTA [24, 44, 47] aims to improve the performance of pre-trained models when they are deployed on unseen domains. TTA strategies are typically bifurcated into two main categories based on their reliance on source data. The first category comprises approaches that adapt models to new domains without direct access to source data [22, 23, 32, 36]. Notable examples within this subset include Tent [47], which updates batch normalization parameters to minimize entropy. Similarly, T3A [15] innovates by creating and updating a set of dynamic prototypes, utilizing them to gauge the similarity between test samples and these prototypes, thereby facilitating adaptation based on feature resemblance.

Conversely, the second category, which our work aligns with, involves techniques that leverage source data to inform the adaptation process. Methods such as NORM [41] and DUA [30] adapt model normalization layers by aligning batch normalization statistics from the training phase with those observed in the target data. Similarly, ActMAD [31] and PromptAlign [40] update all model parameters or employ prompt-based strategies, respectively, utilizing a distribution matching loss that leverages knowledge of source domain statistics. This utilization of source data equips these methods with a foundational understanding of the distribution shift, thereby facilitating more effective adaptation to the target domain.

A significant portion of existing TTA literature centers around Convolutional Neural Networks (CNNs), whose methodologies are not directly transferrable to Vision Transformers (ViTs) due to architectural differences, notably the absence of batch normalization layers in ViTs [15]. This gap underscores the necessity for developing TTA strategies specifically tailored to the unique architecture of ViTs, which our work aims to address.

Innovative approaches like DePT[10], PromptAlign [40], and SwapPrompt [27] have introduced prompt-based strategies for TTA, learning input-specific prompts through test image augmentations. However, these methods encounter limitations due to the restricted number of augmentations available for each test image, struggling to minimize the distance between the comprehensive statistics of the source data and the augmented versions of a single test image. Our methodology circumvents this issue by optimizing a universal prompt across batches, which not only avoids the pitfalls of limited augmentations but also more effectively reduces the gap between source and target domains with only four prompt tokens.

3. Method

In this section, we present OT-VP. We begin with a discussion on the problem setup of TTA in Section 3.1. This is followed by introductions of Vision Transformers in Section 3.2 and Optimal Transport in Section 3.3. Finally, we describe our method in Section 3.4, with the method's workflow illustrated in Figure 2.

3.1. Preliminaries

Problem Definitions. Denote the data from the source (target) domains as $\mathcal{D}^s = \{\mathbf{x}_i^s, y_i^s\}_{i=1}^{n_s}$ ($\mathcal{D}^t = \{\mathbf{x}_i^t, y_i^t\}_{i=1}^{n_t}$), where $\mathbf{x} \in \mathcal{X}$ represents the input image and $y \in \mathcal{Y}$ is the label. The dataset $\mathcal{D}^s$ ($\mathcal{D}^t$) comprises samples that are identically and independently distributed (i.i.d.), characterized by some probability distribution $P^s(X, Y)$ ($P^t(X, Y)$).

In the context of TTA, the model f is initially trained on the source domain, e.g. minimizing the empirical risk,

$$\arg\min_{f} \frac{1}{n_s} \sum_{i=1}^{n_s} \ell(f(\mathbf{x}_i^s), y^s) \tag{1}$$

where ℓ is a loss function. Throughout this paper, we refer to optimizing the model with Eq. 1 as ERM. Generally, the model f is structured as a composition $f = h \circ \phi$, with the feature extractor $\phi : \mathcal{X} \to \mathcal{Z}$ learning the input's representation, and the classifier $h : \mathcal{Z} \to \mathcal{Y}$ predicting the class label.

For any unlabeled target domain $\mathcal{D}^t$, TTA aims to adapt model f or/and target input x^t to bridge the performance gap under the assumption that the source domain and target domain share the same label set. In our approach, we employ a Vision Transformer as the model f, which remains fixed during adaptation.

3.2. Vision Transformers

A Vision Transformer (ViT) [7, 25] processes an input image x by initially dividing it into k patches $\{I_i\}_{i=1}^k$. An encoding layer E is employed to transform the input patches into patch tokens, to which positional embedding are subsequently added to retain spatial information. The inputs to the transformer layers consist of these encoded patch tokens augmented with a special classification token [CLS]. The ViT is composed of several sequential blocks, and each block contains an attention layer and a Multi-Layer Perceptron (MLP) layer. The prediction of the vision transformer can be formulated as follows:

$$\begin{aligned} \texttt{[CLS]} &= \phi([\texttt{[CLS]}, E(I_1), ..., E(I_k)]), \\ y &= h(\texttt{[CLS]}), \end{aligned} \tag{2}$$

where $[\cdot]$ represents concatenation of tokens.

Incorporating a visual prompt into the ViT represents a parameter-efficient approach for fine-tuning or adapting the model, particularly when it is fixed [11, 16]. By introducing l prompt tokens $\{\texttt{[Prompt]}_i\}_{i=1}^l =: \gamma$, the prediction process can be reformulated as follows:

$$\begin{aligned} \texttt{[CLS]} &= \phi([\texttt{[CLS]}, \{E(I_i)\}_{i=1}^k, \gamma]) \\ y &= h(\texttt{[CLS]}) \end{aligned} \tag{3}$$

The optimal prompts can be optimized as follows when the labels are available:

$$\gamma^* = \arg\min_{\gamma} \mathbb{E}[\ell(f(\mathbf{x}; \gamma), y)] \tag{4}$$

3.3. Optimal Transport

Optimal Transport (OT) theory, tracing back to the Monge problem in 1781, evolved significantly with the introduction of the Kantorovich relaxation [17] in 1942. This advancement transformed OT into a robust framework for comparing distributions, shapes, and point clouds [35], leveraging the geometry of the underlying space. OT operates on a complete and separable metric space $\mathcal{X}$, utilizing continuous or discrete probability measures $P, Q \in \mathcal{P}(\mathcal{X})$. The Kantorovich formulation defines the OT problem as:

$$\mathrm{OT}_c(P, Q) := \inf_{\pi \in \Pi(P,Q)} \int_{\mathcal{X} \times \mathcal{X}} c(\mathbf{x}_1, \mathbf{x}_2) d\pi(\mathbf{x}_1, \mathbf{x}_2), \tag{5}$$

where $c(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ denotes a cost function, and $\Pi(P, Q)$ represents the set of all possible couplings or joint distributions over $\mathcal{X} \times \mathcal{X}$ with P and Q as their marginals. The term $W_p(P, Q) := \mathrm{OT}_c(P, Q)^{\frac{1}{p}}$ is referred to as the p-Wasserstein distance when the cost function $c(\mathbf{x}_1, \mathbf{x}_2) = d(\mathbf{x}_1, \mathbf{x}_2)^p$ for some $p \geq 1$ where d is a metric of $\mathcal{X}$.

In real-world applications, the true marginal distributions P, Q are often unknown, leading to reliance on discrete empirical distributions $\hat{P} = \sum_{i=1}^m \mathbf{a}_i \delta_{\mathbf{x}_1^i}$ and $\hat{Q} = \sum_{i=1}^n \mathbf{b}_i \delta_{\mathbf{x}_2^i}$, with $\mathbf{a}, \mathbf{b}$ as vectors in the probability simplex. The cost function then simplifies to an $m \times n$ cost matrix $\mathbf{C}$, where $\mathbf{C}_{ij} = c(\mathbf{x}_1^i, \mathbf{x}_2^j)$. For computational efficiency, the Sinkhorn algorithm [6] introduces an entropic regularizer to the OT problem, facilitating practical applications such as domain adaptation [5] and the evaluation of distances between datasets [1]. This regularized approach, which can be computed using POT [9], allows for the computation of an optimal mapping from source to target domains.

3.4. Test-time Adaptation with OT-VP

In the TTA setting, the absence of labeled target data presents a challenge for prompt optimization as traditionally conducted in Eq. 4. To address this, we introduce an unsupervised prompt adaptation strategy, termed **O**ptimal **T**ransport-guided Test-Time **V**isual **P**rompting (**OT-VP**). This method leverages unlabeled target dataset $\mathcal{D}^t$, passing it through the ViT encoder alongside learnable prompts to obtain a set of representations, as depicted in Figure 2. Source representations, prepared in advance, are readily applied during test time to facilitate OT distance computation.

The essence of OT-VP lies in calculating the OT distance between the source and target representations, underpinned by two distinct cost functions. The first, **OT-VP-base (OT-VP-B)**, measures the cost between two representations devoid of label data, with the cost $c_0(\mathbf{z}^s, \mathbf{z}^t)$ defined as the Euclidean distance between source and target representations $\mathbf{z}^s := h(\mathbf{x}^s)$ and $\mathbf{z}^t := h(\mathbf{x}^s; \gamma)$:

$$c_0(\mathbf{z}^s, \mathbf{z}^t) = \|\mathbf{z}^s - \mathbf{z}^t\|_2 \tag{6}$$

The second cost function, **OT-VP**, enriches this comparison by incorporating label or pseudo-label information, introducing a penalty term scaled by hyperparameter λ for label mismatches between source and target data:

$$c_\lambda((\mathbf{z}^s, y^s), (\mathbf{z}^t, \hat{y}^t)) = \|\mathbf{z}^s - \mathbf{z}^t\|_2 + \lambda \cdot \mathbb{1}_{\{y^s \neq \hat{y}^t\}} \quad (7)$$

where $\hat{y}^t := f(\mathbf{x}^t; \gamma))$ represents the pseudo label derived from the pre-trained model using the adaptively learned prompts. Notably, when setting λ to infinity in Eq. 7, the OT distance is the Wasserstein Distance [43], a well-established mathematical framework that has proven effective for measuring distances between distributions.

The computed OT distance informs the prompt update process for the target dataset, optimizing the prompts to minimize the distance between the source and target distributions. This optimization is formalized as seeking the optimal prompts γ^* that minimize the OT cost, thereby aligning the target dataset's representation with that of the source:

$$\gamma^* = \arg\min_{\gamma} \mathrm{OT}_c(P^s_\#, P^t_\#) \quad (8)$$

where $P^s_\#$ is a joint distribution over source representations and source labels: $(\phi(\mathbf{x}^s), y^s)$, and $P^t_\#$ is a distribution over target representations and target pseudo labels: $(\phi(\mathbf{x}^t), \hat{y}^t)$.

During inference, we apply the optimized prompt tokens γ^* to make predictions for a given target input $\mathbf{x}^t$, following the Eq. 3.

Algorithm	**PACS**	**VLCS**	**OfficeHome**	**Avg**	**Δ Avg**
ERM	64.5$_{\pm 3.2}$	63.8$_{\pm 2.2}$	66.7$_{\pm 0.7}$	65.0	0.0
DoPrompt	64.9$_{\pm 2.3}$	65.8$_{\pm 2.1}$	67.6$_{\pm 0.6}$	65.9	+0.9
Tent-C	64.4$_{\pm 2.4}$	65.0$_{\pm 3.2}$	66.6$_{\pm 1.2}$	66.9	+0.3
Tent-BN	69.0$_{\pm 2.9}$	58.5$_{\pm 2.3}$	67.6$_{\pm 0.9}$	63.5	0.0
T3A	71.2$_{\pm 2.8}$	**68.3**$_{\pm 2.2}$	**68.1**$_{\pm 0.8}$	69.2	+4.2
OT-VP-B	69.8$_{\pm 2.8}$	65.2$_{\pm 1.8}$	66.9$_{\pm 0.5}$	67.3	+2.3
OT-VP	**73.5**$_{\pm 2.7}$	**68.4**$_{\pm 2.3}$	**68.1**$_{\pm 0.7}$	**70.0**	+5.0

Table 1. Accuracy (%) in Single-Source Settings across Datasets and Algorithms. The last column represents the relative improvement over the baseline established by ERM on a ViT-B16 model. Full results can be found in Appendix B.

4. Experiments

4.1. Experimental Setup

Datasets. We evaluate OT-VP on three datasets for TTA: **PACS** [21], **VLCS** [8], and **OfficeHome** [46]. **PACS** [21] is composed of four domains: Photos, Art, Cartoon, and Sketch, containing 9,991 images in 7 classes. **VLCS** [8] comprises four real-world photographic datasets: VOC2007, LabelMe, Caltech, and SUN09, containing 10,729 images in 5 classes. **OfficeHome** [46] consists of four domains: Art, Clipart, Product, Real, containing 15,588 images in 65 classes.

Algorithm	**PACS**	**VLCS**	**OfficeHome**	**Avg**	**Δ Avg**
ERM	87.0$_{\pm 0.1}$	78.5$_{\pm 0.3}$	73.6$_{\pm 0.1}$	79.7	0.0
DoPrompt	**87.5**$_{\pm 0.2}$	80.3$_{\pm 0.4}$	74.7$_{\pm 0.2}$	80.8	+1.1
Tent-C	87.2$_{\pm 0.2}$	78.8$_{\pm 0.5}$	73.6$_{\pm 0.2}$	79.9	+0.2
Tent-BN	87.2$_{\pm 0.2}$	78.5$_{\pm 0.3}$	74.6$_{\pm 0.3}$	78.4	+0.4
T3A	87.4$_{\pm 0.1}$	80.0$_{\pm 0.4}$	74.7$_{\pm 0.2}$	80.7	+1.0
OT-VP-B	87.3$_{\pm 0.2}$	80.2$_{\pm 0.3}$	74.3$_{\pm 0.3}$	80.6	+0.9
OT-VP	**87.7**$_{\pm 0.1}$	**80.9**$_{\pm 0.4}$	**75.1**$_{\pm 0.1}$	**81.2**	+1.5

Table 2. Accuracy (%) in Multi-Source Settings across Datasets and Algorithms. The last column represents the relative improvement over the baseline established by ERM on a ViT-B16 model. Full results can be found in Appendix B.

Baselines. We compare OT-VP against the following state-of-the-art TTA approaches:

1. Tent [47] is a TTA method that fine-tunes the batch normalization (BN) parameters to minimize prediction entropy during test time. Notably, ViTs do not contain BN layers. Following the methodology outlined by [15], we adapt Tent for use with ViTs in two distinct manners. Specifically, Tent-BN introduces a BN layer immediately preceding the linear classifier, allowing for the adjustment of normalization parameters within this BN layer. Tent-C modulates the entire classifier to reduce prediction entropy.
2. T3A [15] begins with generating pseudo-prototype representations for each class using unlabeled target data and the pre-trained classifier. Subsequently, the classification of each target sample is performed based on its distance to the pseudo-prototypes.
3. DoPrompt [49] learns domain-specific prompts for each source domain during the training phase. Additionally, it employs a prompt adapter, a mechanism trained to craft an appropriate prompt for individual input images. At test time, this prompt adapter generates an input-specific prompt based on the learned source domain prompts. Note that DoPrompt is NOT a TTA method as the prompts are optimized during the training time.

Implementation details. Following [13], we partition the data from each domain into training and validation splits of 80% and 20%, respectively, utilizing the larger split for training and the smaller one for model selection. For all experiments, we utilize the ViT-B16 model pre-trained on ImageNet, sourced from PyTorch [34]. Our training approach for ERM adheres to the hyperparameters specified by [49], incorporating a dropout rate of 0.1 and a weight decay of 10^{-2}. The learning rate is 5×10^{-6} for PACS and VLCS, and 10^{-5} for OfficeHome.

For all baseline methods, we use their official implementation[1] [2] [3] (see details in Appendix B.2). For implementation of our methods, we uses the `DomainBed` library [13] [4]. Adopting DoPrompt's strategy, we use 4 prompts, optimized via the AdamW optimizer [18, 26] with a 0.1 learning rate on the target validation set. The target dataset's batch size is 64, with prompt training capped at 5 epochs to avoid overfitting. We set λ in Eq. 6 to 10^8, as an estimate of infinity. All experiments were conducted using an RTX 6000 Ada GPU.

4.2. Results

Tables 1 and 2 summarize our experimental outcomes in single-source and multi-source settings, respectively. In the single-source scenario, the model is trained on one domain and then adapted to another. The average accuracy is calculated across all 12 domain pairings for each trial. In the multi-source setting, one domain is designated as the target while the remaining three serve as sources. Each reported value is an average, with standard deviations across three separate trials. These trials vary by weight initialization and data splits. Comprehensive results can be found in the Appendix B.

OT-VP consistently outperforms SOTA TTA methods across all datasets in both settings. Demonstrated by the results in Tables 1 and 2, OT-VP remarkably boosts the ERM model's performance in both single-source and multi-source settings. Notably, OT-VP secures substantial improvements, with enhancements of 9.0%, 4.6%, and 1.4% in the single-source scenario, and 0.7%, 2.4%, and 1.5% in the multi-source scenario across the respective datasets: PACS, VLSC, and OfficeHome, highlighting its potent effect on model accuracy.

The contrast in performance between alternative approaches and our OT-VP is noteworthy. The observation that Tent-BN and Tent-C exhibit only nominal or no improvements, and sometimes even lead to declines in performance on ViT models, aligns with expectations considering the original design of Tent. Tent was initially conceived for models relying heavily on batch normalization layers for domain adaptation by adjusting these layers to minimize prediction entropy. However, ViTs operate on a different architectural principle, lacking batch normalization layers in their standard configuration. This fundamental discrepancy underscores the challenges of directly applying methodologies designed for CNNs to ViTs without modifications or considerations of architectural differences. Furthermore, T3A consistently improves performance over Tent in both experimental setups, though its gains are lower than those achieved by OT-VP.

Remarkably, OT-VP demonstrates significant potential in the single-source setting. In such cases, while methods like T3A show limited improvements, OT-VP significantly boosts accuracy. For example, in the PACS dataset with P as the source and S as the target ($P \rightarrow S$), T3A increases accuracy by only 1.8%. In contrast, OT-VP raises it by 33.1%. Another case from A to C ($A \rightarrow C$) in PACS sees OT-VP improving accuracy by 17.3%, against T3A's 5.7%. These results are depicted in a t-SNE visualization 3. For more details, see the Appendix, Table 4.

Moreover, while ERM on ViTs demonstrates significant potential in adapting to unseen target domains, many prior DG and TTA methods struggle to enhance performance [15, 49]. This highlights the superior adaptability and efficacy of OT-VP.

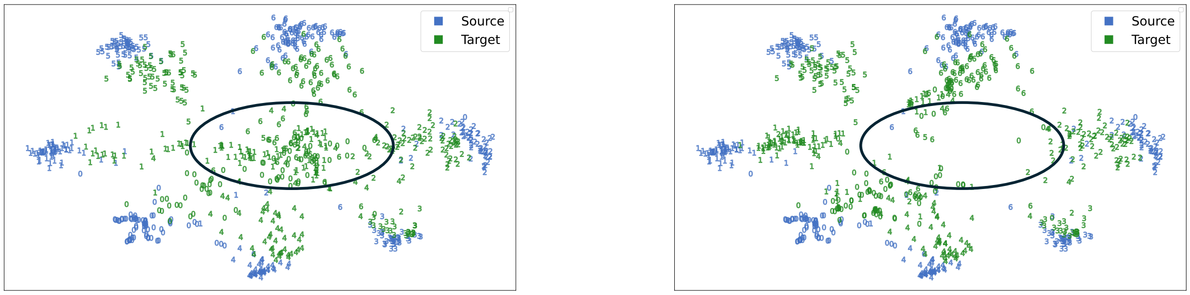
OT-VP achieves superior performance over training-time prompt learning. Unlike DoPrompt, our approach eliminates the need to delve into the training process, significantly enhancing practical applicability. By leveraging target data, OT-VP adeptly navigates the distributional shifts present in the target domain. It optimizes target-specific prompts to effectively bridge any gaps, moving beyond reliance solely on source domain insights.

DoPrompt, while innovative, learns domain prompts by treating training data as out-of-distribution (OOD) samples. This strategy does not ensure that the prompts will perform well across all potential target domains. Furthermore, DoPrompt requires prompt learning for each domain alongside a prompt adapter (an auxiliary model) to tailor prompts for every target input image. In contrast, our method optimizes universal prompts for a singular target domain, streamlining the process.

As evidenced in Table 1, DoPrompt experiences significant performance drops when limited to a single domain, which could limit its applicability in scenarios where source data is scarce, such as in medical imaging. Our method surpasses DoPrompt in both scenarios, underscoring the efficiency and accessibility of learning prompts at test time.

OT-VP implicitly reduces prediction entropy. Consistent with prior research [47], there's an observed correlation between prediction entropy and accuracy—lower entropy often signifies more accurate and confident predictions. Unlike traditional approaches that explicitly target entropy reduction by adjusting model parameters [40, 47], OT-VP achieves this indirectly through the strategic application of Optimal Transport. This involves leveraging a cost metric that encompasses both features and labels 7, aiming to align the target distribution more closely with the source

[1] https://github.com/DequanWang/tent
[2] https://github.com/matsuolab/T3A
[3] https://github.com/zhengzangw/DoPrompt
[4] https://github.com/facebookresearch/DomainBed



(a) Before OT-VP

(b) After OT-VP

Figure 3. t-SNE visualization showcasing the impact of OT-VP. The figures display the representation space before and after the application of OT-VP for A → C in the PACS dataset with the pre-trained ViT encoder. Different numbers represent distinct class labels. (a) The initial state from ERM, as indicated in the left image, shows the target data points are not only distant from the source but also exhibit considerable class overlap, especially within the central region enclosed by the ellipse. This misalignment reflects an accuracy of 63.5%, an OT distance of 29.1, and a prediction entropy of 0.54. (b) After employing OT-VP, the right image shows that the target representations become more distinct and well-separated, with classes from source and target better aligned. The target data have shifted closer to the corresponding source representations, improving accuracy to 81.4%—an increase of **17.9%**, and reducing the OT distance and prediction entropy to 25.8 and 0.27 respectively.

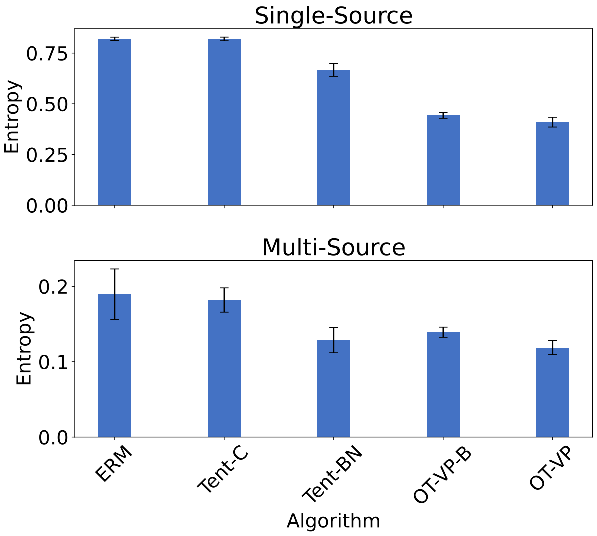


Figure 4. Prediction entropy across Algorithms in Single-Source and Multi-Source settings on PACS. In both settings, OT-VP demonstrates a marked reduction in entropy, outperforming Tent-C and Tent-BN, which target entropy minimization directly.

distribution, thereby enhancing model confidence near the decision boundary. This alignment is visually supported by representations such as those depicted in Fig. 3, a t-SNE visualization for source A and the target C (A → C) within the PACS dataset.

A comparative analysis of prediction entropy among ERM, Tent-C, Tent-BN, and OT-VP—illustrated in Fig. 4—demonstrates that OT-VP can significantly lower entropy through the refined optimization of prompts. Remarkably, it does so even when compared with methods like Tent-C and Tent-BN, which pursue entropy minimization directly. It's important to note that the improvements achieved by Tent-C and Tent-BN result from carefully balancing accuracy and entropy reduction when selecting their hyperparameters.

4.3. Ablation Study

Analysis on the objective functions. To explore the efficacy of our approach, we experimented with directly minimizing entropy during the prompt learning process. Interestingly, this direct focus on entropy reduction did not yield improvements. Instead, we observed a monotonic decrease in accuracy as entropy was reduced. This suggests that while reducing entropy might intuitively seem beneficial, focusing solely on this aspect can lead to overconfidence in the model's predictions, including those that are incorrect. Such over-trust manifests as high confidence in erroneous predictions, underlining a potential pitfall of learning universal visual prompts purely on entropy reduction. These findings validate the effectiveness of our chosen objective function, which not only achieves the desired outcome but also implicitly manages entropy without compromising the reliability of predictions.

Analysis on λ. In our further investigations, we delved into the impact of the hyperparameter λ within Eq. 7, particularly in the single-source setting for PACS, across two distinct scenarios: S→C, P→S, and P→C. These scenarios represent conditions where OT-VP underperforms, outperforms, and performs similarly to other methods, respectively. Our experiments spanned a range of λ values:

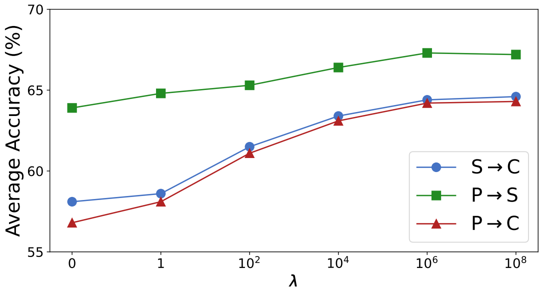


Figure 5. Influence of hyperparameter λ on Single-Source performance. Accuracy declines with smaller λ values but stabilizes when λ is large. These trends reveal the pivotal role of λ in preventing cross-class transport and its impact on overfitting, particularly when using pseudo labels during prompt optimization. Notably, OT-VP's λ is 10^8.

$[0, 1, 10^2, 10^4, 10^6, 10^8]$, where $\lambda = 0$ and $\lambda = 10^8$ corresponds to the OT-VP-B and OT-VP, respectively (details in Appendix 4).

Our results highlight two crucial insights: Firstly, incorporating label information into the learned prompts significantly enhances the alignment between the source and target domains, corroborating findings from prior research [1]. This label augmentation facilitates a deeper, more meaningful adaptation process. Secondly, we observed a performance decrement with the reduction of λ values, as illustrated in Fig. 5. However, performance reaches a plateau and stabilizes when λ is sufficiently large, approaching an effective infinity in relation to feature distance, indicating an optimal balance in the adaptation mechanism.

The role of λ in the cost function is crucial—it modulates the extent to which transportation across different classes is permissible. A higher λ effectively restricts cross-class transport, serving as a preventative measure against such occurrences. Notably, setting λ to infinity aligns the OT distance with the Wasserstein distance, underlining the significance of this parameter. Our analysis suggests that a larger λ facilitates a reduction in overfitting risk during prompt optimization with stochastic gradient descent (SGD), especially when utilizing pseudo labels. This is achieved by imposing a stricter barrier against cross-class transportation, thereby ensuring a more disciplined adaptation process.

Analysis on prompt length. We opted for a prompt length of 4, aligning with DoPrompt's configuration. Our experiments across varying prompt lengths—2, 4, 6, and 8—revealed that OT-VP's performance remained relatively stable for smaller lengths (2, 4, 6), as indicated in Fig. 6. However, a notable decline in performance was observed with a length of 8, attributed to the increased risk of overfitting, a consequence of the absence of target labels. The decision to fix the prompt length is driven by the necessity to pre-select all parameters prior to any engagement with the target data, ensuring a standardized approach in our experiments.

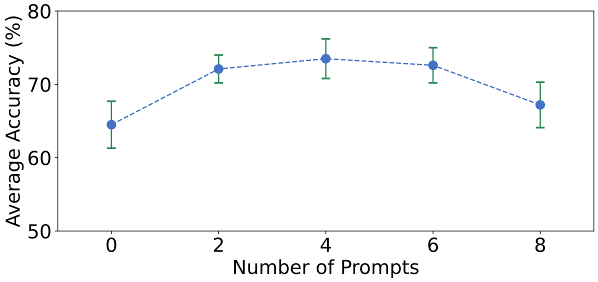


Figure 6. Effect of Prompt Length on OT-VP Performance. Consistency in performance is maintained for shorter prompt lengths (2, 4, 6), but an extended length of 8 prompts leads to a performance drop, likely due to overfitting in the absence of target labels.

Analysis on computation time. Despite employing SGD to optimize prompt tokens, OT-VP remains computationally efficient. We optimize only 4 prompt tokens over 5 epochs, utilizing a 20% hold-out split from both source and target data. While the multi-source setting involves processing triple the data to compute source representations compared to the single-source setting, the time required for both is nearly identical. Specifically, the average time is 39.5 seconds for the multi-source and 38.7 seconds for the single-source setting on the PACS dataset on our hardware. Moreover, the computational time is slightly influenced by the size of the datasets but remains relatively quick. For instance, in the PACS dataset, domain S, which has more than double the data of domain P, requires more processing time—51.7 seconds for S versus 34.5 seconds for P in the multi-source setting. Full results for PACS can be found in Table 3 in Appendix. In conclusion, OT-VP can efficiently learn prompts in both single and multi-source settings without significant computational overhead.

5. Conclusion

In this paper, we present OT-VP, a novel test-time adaptation approach that leverages visual prompt learning for effective test-time adaptation. OT-VP stands out by adapting without altering the pre-trained model and effectively aligning the source and target domains, offering a more practical solution than established methods. Our experiments reveal OT-VP's strengths: consistently outperforming existing TTA methods for ViT and DG prompt learning, and reducing prediction entropy to increase model confidence. By optimizing universal prompts for the target domain, OT-VP simplifies the adaptation process, enhancing the applicability of deep learning models in real-world settings.

References

[1] David Alvarez-Melis and Nicolo Fusi. Geometric dataset distances via optimal transport. *Advances in Neural Information Processing Systems*, 33:21428–21439, 2020. 4, 8

[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*, 2(3):7, 2022. 3

[3] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems*. MIT Press, 2006. 2

[4] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. *Advances in neural information processing systems*, 24, 2011. 1

[5] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865, 2016. 4

[6] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013. 4

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 1, 3, 4

[8] Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1657–1664, 2013. 5

[9] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. 4

[10] Yunhe Gao, Xingjian Shi, Yi Zhu, Hao Wang, Zhiqiang Tang, Xiong Zhou, Mu Li, and Dimitris N. Metaxas. Visual prompt tuning for test-time domain adaptation, 2023. 2, 3

[11] Chunjiang Ge, Rui Huang, Mixue Xie, Zihang Lai, Shiji Song, Shuang Li, and Gao Huang. Domain adaptation via prompt learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023. 4

[12] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012. 2

[13] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020. 5, 6

[14] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 1

[15] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. In *Advances in Neural Information Processing Systems*, pages 2427–2440. Curran Associates, Inc., 2021. 1, 2, 3, 5, 6

[16] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer, 2022. 1, 3, 4

[17] Leonid V Kantorovich. On the translocation of masses. In *Dokl. Akad. Nauk. USSR (NS)*, pages 199–201, 1942. 2, 4

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[19] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pages 5637–5664. PMLR, 2021. 1

[20] Trung Le, Tuan Nguyen, Nhat Ho, Hung Bui, and Dinh Phung. Lamda: Label matching deep domain adaptation. In *Proceedings of the 38th International Conference on Machine Learning*, pages 6043–6054. PMLR, 2021. 2

[21] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017. 1, 5

[22] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International conference on machine learning*, pages 6028–6039. PMLR, 2020. 3

[23] Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8602–8617, 2021. 3

[24] Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *arXiv preprint arXiv:2303.15361*, 2023. 3

[25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 1, 3, 4

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[27] Xiaosong Ma, Jie Zhang, Song Guo, and Wenchao Xu. Swapprompt: Test-time prompt adaptation for vision-language models. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 3

[28] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009. 2

[29] Akshay Mehra, Yunbei Zhang, and Jihun Hamm. Analysis of task transferability in large pre-trained classifiers. *arXiv preprint arXiv:2307.00823*, 2023. 2

[30] M Jehanzeb Mirza, Jakub Micorek, Horst Possegger, and Horst Bischof. The norm must go on: Dynamic unsupervised domain adaptation by normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14765–14775, 2022. 3

[31] Muhammad Jehanzeb Mirza, Pol Jané Soneira, Wei Lin, Mateusz Kozinski, Horst Possegger, and Horst Bischof. Actmad: Activation matching to align distributions for test-time-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24152–24161, 2023. 3

[32] Chaithanya Kumar Mummadi, Robin Hutmacher, Kilian Rambach, Evgeny Levinkov, Thomas Brox, and Jan Hendrik Metzen. Test-time adaptation to distribution shift by confidence maximization and input transformation. *arXiv preprint arXiv:2106.14999*, 2021. 3

[33] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34: 23296–23308, 2021. 1

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 5

[35] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019. 2, 4

[36] Mihir Prabhudesai, Tsung-Wei Ke, Alexander Cong Li, Deepak Pathak, and Katerina Fragkiadaki. Diffusion-tta: Test-time adaptation of discriminative models via generative feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 3

[37] 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

[38] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in neural information processing systems*, 34:12116–12128, 2021. 1

[39] 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. 1

[40] 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, 3, 6

[41] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. *Advances in neural information processing systems*, 33:11539–11551, 2020. 3

[42] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 2

[43] Aman Sinha, Hongseok Namkoong, Riccardo Volpi, and John Duchi. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017. 5

[44] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pages 9229–9248. PMLR, 2020. 3

[45] Yang Tan, Yang Li, and Shao-Lun Huang. Otce: A transferability metric for cross-domain cross-task representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15779–15788, 2021. 2

[46] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

[47] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. 1, 3, 5, 6

[48] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022. 3

[49] Zangwei Zheng, Xiangyu Yue, Kai Wang, and Yang You. Prompt vision transformer for domain generalization. *arXiv preprint arXiv:2208.08914*, 2022. 1, 2, 3, 5, 6

[50] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4396–4415, 2022. 1

OT-VP: Optimal Transport-guided Visual Prompting for Test-Time Adaptation

Supplementary Material

A. Limitations

One limitation of our OT-VP approach is its reliance on the quality of pseudo labels for computing the Optimal Transport distance. As visualized in our t-SNE plots 3, there's a risk of occasional misalignment due to inaccurate pseudo-labeling, which can adversely affect the model's ability to accurately bridge the source and target domain gap. While implementing entropy-based filtering akin to T3A [15] could mitigate this by filtering out high-entropy, less reliable pseudo labels, the fundamental limitation remains: OT-VP's capacity to perform effective test-time adaptation may be significantly hindered if the pseudo labels are entirely unreliable or carry no meaningful information about the true class distribution.

B. Full Results

B.1.

In this section, we present the computation time for OT-VP on the PACS dataset. For the single-source setting, the average computation time is calculated across three different sources.

Setting	A	C	P	S	Average
Single-Source	32.6	37.5	33.7	50.8	38.7
Multi-Source	33.9	38.0	34.5	51.7	39.5

Table 3. Average computation time (seconds) for OT-VP on PACS dataset.

B.2.

We present the implementation details of all baseline methods. For the implementation of DoPrompt, we set the prompt length to 4, with the coefficient λ explored over the set $\{0.1, 1, 10\}$. The M parameter for T3A is chosen from $\{1, 5, 20, 50, 100, \mathrm{N/A}\}$, while the configuration for Tent is determined from combinations of $\{0.1, 1.0, 10.0\}$ and $\{1, 3\}$.

B.3.

In this section, we present the comprehensive outcomes in Tables 1 and 2. The experiments were conducted using three different seeds {0, 1, 2} within the `DomainBed` framework.

Algorithm		A	C	P	S
ERM	**A**	-	$64.5_{\pm0.9}$	$98.9_{\pm0.2}$	$56.4_{\pm2.0}$
	C	$83.9_{\pm3.3}$	-	$89.6_{\pm0.5}$	$69.2_{\pm2.1}$
	P	$74.2_{\pm1.4}$	$44.4_{\pm1.2}$	-	$34.1_{\pm4.3}$
	S	$50.8_{\pm3.6}$	$58.4_{\pm6.0}$	$49.5_{\pm4.4}$	-
DoPrompt	**A**	-	$64.6_{\pm0.6}$	$98.5_{\pm0.7}$	$56.5_{\pm4.7}$
	C	$84.1_{\pm2.2}$	-	$90.1_{\pm1.7}$	$74.0_{\pm2.2}$
	P	$75.6_{\pm2.0}$	$46.2_{\pm0.7}$	-	$35.2_{\pm3.8}$
	S	$46.4_{\pm4.9}$	$55.0_{\pm5.8}$	$45.1_{\pm3.0}$	-
Tent-C	**A**	-	$64.6_{\pm0.6}$	$98.9_{\pm0.2}$	$56.3_{\pm5.8}$
	C	$83.9_{\pm3.0}$	-	$89.6_{\pm0.2}$	$69.0_{\pm2.0}$
	P	$74.4_{\pm1.9}$	$44.5_{\pm1.3}$	-	$33.7_{\pm4.9}$
	S	$50.4_{\pm3.5}$	$58.3_{\pm6.0}$	$49.0_{\pm3.7}$	-
Tent-BN	**A**	-	$71.9_{\pm0.9}$	$98.9_{\pm0.5}$	$66.6_{\pm3.7}$
	C	$84.9_{\pm2.1}$	-	$91.3_{\pm1.7}$	$71.7_{\pm1.0}$
	P	$78.0_{\pm1.7}$	$56.0_{\pm3.6}$	-	$41.8_{\pm2.5}$
	S	$56.3_{\pm4.0}$	$62.9_{\pm3.1}$	$47.5_{\pm2.9}$	-
T3A	**A**	-	$70.2_{\pm2.1}$	$98.6_{\pm0.5}$	$67.9_{\pm4.5}$
	C	$86.3_{\pm1.7}$	-	$94.4_{\pm1.4}$	$71.1_{\pm1.4}$
	P	$80.2_{\pm1.6}$	$53.9_{\pm3.9}$	-	$35.9_{\pm2.4}$
	S	$69.0_{\pm4.0}$	$69.9_{\pm5.9}$	$56.9_{\pm2.5}$	-
OT-VP-B	**A**	-	$76.7_{\pm2.9}$	$98.3_{\pm0.1}$	$66.8_{\pm2.8}$
	C	$84.4_{\pm0.8}$	-	$92.2_{\pm2.2}$	$69.8_{\pm1.2}$
	P	$77.8_{\pm0.4}$	$56.8_{\pm6.9}$	-	$63.9_{\pm2.1}$
	S	$44.7_{\pm3.1}$	$58.1_{\pm3.6}$	$40.4_{\pm3.3}$	-
OT-VP	**A**	-	$81.8_{\pm0.3}$	$99.0_{\pm0.3}$	$72.2_{\pm2.9}$
	C	$84.4_{\pm2.2}$	-	$92.6_{\pm1.6}$	$69.5_{\pm0.8}$
	P	$80.4_{\pm1.7}$	$64.3_{\pm4.2}$	-	$67.2_{\pm2.6}$
	S	$56.0_{\pm3.9}$	$64.6_{\pm4.4}$	$50.5_{\pm1.0}$	-

Table 4. Single-Source Full Results on PACS in Table 1

Algorithm	A	C	P	S	Average
ERM	$91.3_{\pm0.5}$	$82.3_{\pm0.7}$	$98.9_{\pm0.3}$	$75.6_{\pm0.4}$	87.0
DoPrompt	**91.4**$_{\pm0.5}$	$81.8_{\pm0.4}$	**99.5**$_{\pm0.4}$	**77.1**$_{\pm0.5}$	87.5
Tent-C	**91.6**$_{\pm0.7}$	**82.7**$_{\pm1.3}$	$98.9_{\pm0.4}$	**75.7**$_{\pm0.5}$	87.2
Tent-BN	$91.1_{\pm0.3}$	**82.4**$_{\pm1.3}$	$98.3_{\pm0.4}$	**76.8**$_{\pm0.7}$	87.2
T3A	**91.5**$_{\pm0.5}$	$81.8_{\pm0.9}$	**99.0**$_{\pm0.5}$	**77.4**$_{\pm0.3}$	87.4
OT-VP-B	**91.2**$_{\pm0.5}$	$81.8_{\pm0.7}$	**99.4**$_{\pm0.5}$	**77.4**$_{\pm0.5}$	87.3
OT-VP	**92.0**$_{\pm0.5}$	**83.0**$_{\pm0.4}$	**99.2**$_{\pm0.4}$	**76.4**$_{\pm0.2}$	87.7

Table 5. Multi-Source Full Results on PACS in Table 2

Algorithm		C	L	S	V
ERM	**C**	-	50.7$_{\pm 1.2}$	47.9$_{\pm 2.3}$	47.0$_{\pm 2.1}$
	L	62.9$_{\pm 1.4}$	-	55.8$_{\pm 1.6}$	63.1$_{\pm 2.2}$
	S	67.5$_{\pm 1.5}$	59.9$_{\pm 1.3}$	-	67.7$_{\pm 2.4}$
	V	96.5$_{\pm 0.7}$	66.1$_{\pm 2.1}$	80.3$_{\pm 1.5}$	-
DoPrompt	**C**	-	53.4$_{\pm 1.5}$	50.0$_{\pm 1.6}$	50.5$_{\pm 1.8}$
	L	71.7$_{\pm 2.1}$	-	57.8$_{\pm 1.5}$	70.1$_{\pm 1.3}$
	S	67.8$_{\pm 1.9}$	62.5$_{\pm 0.8}$	-	66.2$_{\pm 1.9}$
	V	98.6$_{\pm 0.8}$	62.0$_{\pm 1.9}$	78.8$_{\pm 2.1}$	-
Tent-C	**C**	-	50.4$_{\pm 1.7}$	48.3$_{\pm 1.3}$	47.0$_{\pm 2.9}$
	L	70.3$_{\pm 1.9}$	-	55.8$_{\pm 2.4}$	63.2$_{\pm 2.1}$
	S	67.2$_{\pm 2.1}$	59.8$_{\pm 1.7}$	-	67.9$_{\pm 2.0}$
	V	96.5$_{\pm 0.6}$	66.0$_{\pm 2.1}$	88.2$_{\pm 1.8}$	-
Tent-BN	**C**	-	71.9$_{\pm 2.6}$	98.9$_{\pm 2.1}$	66.6$_{\pm 1.8}$
	L	84.9$_{\pm 2.8}$	-	91.3$_{\pm 3.0}$	71.7$_{\pm 2.1}$
	S	78.0$_{\pm 1.9}$	56.0$_{\pm 0.7}$	-	41.8$_{\pm 1.8}$
	V	56.3$_{\pm 2.2}$	62.9$_{\pm 1.9}$	47.5$_{\pm 1.3}$	-
T3A	**C**	-	51.8$_{\pm 1.6}$	57.0$_{\pm 1.8}$	57.3$_{\pm 1.2}$
	L	83.6$_{\pm 1.8}$	-	62.7$_{\pm 1.2}$	64.3$_{\pm 2.1}$
	S	71.1$_{\pm 1.9}$	60.5$_{\pm 1.7}$	-	67.4$_{\pm 1.8}$
	V	97.3$_{\pm 0.6}$	66.8$_{\pm 1.9}$	80.3$_{\pm 1.4}$	-
OT-VP-B	**C**	-	55.7$_{\pm 2.4}$	50.0$_{\pm 2.1}$	47.0$_{\pm 2.6}$
	L	73.1$_{\pm 1.8}$	-	56.4$_{\pm 1.8}$	60.7$_{\pm 2.1}$
	S	67.1$_{\pm 2.3}$	60.8$_{\pm 1.5}$	-	67.3$_{\pm 1.9}$
	V	96.8$_{\pm 0.6}$	68.4$_{\pm 1.9}$	79.1$_{\pm 1.2}$	-
OT-VP	**C**	-	59.9$_{\pm 1.2}$	51.3$_{\pm 2.3}$	48.9$_{\pm 0.8}$
	L	90.8$_{\pm 0.6}$	-	56.3$_{\pm 1.2}$	63.8$_{\pm 0.9}$
	S	69.6$_{\pm 1.9}$	64.2$_{\pm 0.8}$	-	68.8$_{\pm 1.2}$
	V	96.8$_{\pm 0.6}$	69.3$_{\pm 0.9}$	80.8$_{\pm 1.5}$	-

Table 6. Single-Source Full Results on VLCS in Table 1

Algorithm	C	L	S	V	Average
ERM	96.5$_{\pm 0.5}$	65.5$_{\pm 1.1}$	75.2$_{\pm 0.8}$	76.7$_{\pm 0.4}$	78.5
DoPrompt	**98.2**$_{\pm 0.8}$	**67.8**$_{\pm 0.8}$	75.3$_{\pm 0.7}$	**79.9**$_{\pm 0.5}$	80.3
Tent-C	**97.7**$_{\pm 0.5}$	65.2$_{\pm 1.0}$	75.3$_{\pm 0.7}$	76.9$_{\pm 0.4}$	78.8
Tent-BN	86.3$_{\pm 1.0}$	**66.2**$_{\pm 0.9}$	68.8$_{\pm 0.7}$	72.6$_{\pm 0.5}$	73.5
T3A	**97.3**$_{\pm 0.5}$	65.6$_{\pm 0.9}$	**78.0**$_{\pm 0.7}$	**79.3**$_{\pm 0.7}$	80.0
OT-VP-B	**96.8**$_{\pm 0.4}$	**71.9**$_{\pm 0.8}$	75.2$_{\pm 0.8}$	76.9$_{\pm 0.4}$	80.2
OT-VP	**96.8**$_{\pm 0.4}$	**73.1**$_{\pm 1.1}$	**76.8**$_{\pm 0.8}$	**77.0**$_{\pm 0.5}$	80.9

Table 7. Multi-Source Full Results on VLCS in Table 2

Algorithm		A	C	P	R
ERM	**A**	-	54.3$_{\pm 0.5}$	71.4$_{\pm 1.6}$	77.0$_{\pm 0.4}$
	C	67.4$_{\pm 0.7}$	-	70.0$_{\pm 1.2}$	73.2$_{\pm 0.4}$
	P	62.9$_{\pm 2.3}$	47.8$_{\pm 0.1}$	-	78.9$_{\pm 0.7}$
	R	70.3$_{\pm 0.7}$	49.2$_{\pm 1.8}$	78.5$_{\pm 0.5}$	-
DoPrompt	**A**	-	52.1$_{\pm 3.2}$	71.7$_{\pm 0.1}$	79.0$_{\pm 0.5}$
	C	67.4$_{\pm 0.1}$	-	71.7$_{\pm 0.4}$	75.5$_{\pm 0.1}$
	P	66.8$_{\pm 1.4}$	47.8$_{\pm 0.1}$	-	79.0$_{\pm 0.1}$
	R	72.2$_{\pm 1.1}$	48.8$_{\pm 1.9}$	79.6$_{\pm 0.7}$	-
Tent-C	**A**	-	54.5$_{\pm 0.6}$	69.9$_{\pm 1.5}$	76.9$_{\pm 0.6}$
	C	66.9$_{\pm 0.7}$	-	69.6$_{\pm 1.4}$	73.6$_{\pm 0.2}$
	P	62.9$_{\pm 0.3}$	47.9$_{\pm 0.1}$	-	78.6$_{\pm 0.7}$
	R	71.0$_{\pm 0.5}$	47.1$_{\pm 1.7}$	79.9$_{\pm 0.3}$	-
Tent-BN	**A**	-	56.7$_{\pm 0.7}$	70.5$_{\pm 0.8}$	77.5$_{\pm 0.8}$
	C	67.9$_{\pm 0.6}$	-	70.4$_{\pm 1.4}$	72.9$_{\pm 0.4}$
	P	65.4$_{\pm 1.2}$	48.6$_{\pm 0.3}$	-	79.1$_{\pm 0.3}$
	R	72.8$_{\pm 0.5}$	49.5$_{\pm 1.2}$	79.9$_{\pm 0.3}$	-
T3A	**A**	-	55.1$_{\pm 0.7}$	71.2$_{\pm 1.2}$	76.6$_{\pm 0.7}$
	C	67.9$_{\pm 0.7}$	-	71.5$_{\pm 1.2}$	74.8$_{\pm 0.3}$
	P	67.1$_{\pm 0.8}$	48.7$_{\pm 0.2}$	-	80.3$_{\pm 0.5}$
	R	72.9$_{\pm 0.5}$	49.8$_{\pm 1.5}$	80.9$_{\pm 0.4}$	-
OT-VP-B	**A**	-	55.1$_{\pm 0.8}$	70.5$_{\pm 0.8}$	75.0$_{\pm 0.5}$
	C	65.8$_{\pm 0.4}$	-	69.4$_{\pm 0.8}$	73.1$_{\pm 0.2}$
	P	64.6$_{\pm 1.8}$	49.1$_{\pm 0.1}$	-	77.4$_{\pm 0.4}$
	R	71.1$_{\pm 0.9}$	52.1$_{\pm 1.8}$	79.6$_{\pm 0.6}$	-
OT-VP	**A**	-	55.0$_{\pm 0.7}$	71.4$_{\pm 1.2}$	76.9$_{\pm 0.7}$
	C	67.6$_{\pm 0.7}$	-	70.1$_{\pm 0.9}$	73.6$_{\pm 0.8}$
	P	68.7$_{\pm 0.9}$	49.7$_{\pm 0.6}$	-	79.9$_{\pm 0.5}$
	R	71.3$_{\pm 0.5}$	52.2$_{\pm 1.3}$	80.8$_{\pm 0.4}$	-

Table 8. Single-Source Full Results on OfficeHome in Table 1

Algorithm	A	C	P	S	Average
ERM	73.8$_{\pm 0.2}$	57.3$_{\pm 0.6}$	80.3$_{\pm 0.1}$	83.0$_{\pm 0.8}$	73.6
DoPrompt	73.4$_{\pm 0.4}$	**58.8**$_{\pm 0.5}$	**81.7**$_{\pm 0.2}$	**84.8**$_{\pm 0.7}$	74.7
Tent-C	73.5$_{\pm 0.4}$	57.3$_{\pm 0.6}$	**80.4**$_{\pm 0.1}$	**83.2**$_{\pm 0.9}$	73.6
Tent-BN	73.5$_{\pm 0.4}$	**58.8**$_{\pm 0.5}$	**81.9**$_{\pm 0.2}$	**84.0**$_{\pm 0.8}$	74.6
T3A	**74.2**$_{\pm 0.3}$	**58.3**$_{\pm 0.5}$	**81.8**$_{\pm 0.1}$	**84.6**$_{\pm 0.7}$	74.7
OT-VP-B	**74.0**$_{\pm 0.4}$	**58.9**$_{\pm 0.5}$	**80.7**$_{\pm 0.2}$	**83.5**$_{\pm 0.7}$	74.3
OT-VP	**74.2**$_{\pm 0.4}$	**59.6**$_{\pm 0.5}$	**82.3**$_{\pm 0.1}$	**84.1**$_{\pm 0.8}$	75.1

Table 9. Multi-Source Full Results on OfficeHome in Table 2