VISUAL PROMPT TUNING FOR TEST-TIME DOMAIN ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Models should have the ability to adapt to unseen data during test-time to avoid performance drop caused by inevitable distribution shifts in real-world deployment scenarios. In this work, we tackle the practical yet challenging test-time adaptation (TTA) problem, where a model adapts to the target domain without accessing the source data. We propose a simple recipe called data-efficient prompt tuning (DePT) with two key ingredients. First, DePT plugs visual prompts into the vision Transformer and only tunes these source-initialized prompts during adaptation. We find such parameter-efficient finetuning can efficiently adapt the model representation to the target domain without overfitting to the noise in the learning objective. Second, DePT bootstraps the source representation to the target domain by memory bank-based online pseudo labeling. A hierarchical self-supervised regularization specially designed for prompts is jointly optimized to alleviate error accumulation during self-training. With much fewer tunable parameters, DePT demonstrates not only state-of-the-art performance on major adaptation benchmarks, but also superior data efficiency, i.e., adaptation with only 1% or 10% data without much performance degradation compared to 100% data. In addition, DePT is also versatile to be extended to online or multi-source TTA settings.

1 INTRODUCTION

Deep neural networks achieve excellent performance when the testing data (target domain) follow the same distribution as the training data (source domain). However, their generalization ability on the testing data is not guaranteed when there is a distribution shift between the source and the target. Even simple domain shift, like common corruptions (Hendrycks & Dietterich, 2019) or appearance variations (Geirhos et al., 2018), can lead to significant performance drop. Solving the problem of domain shift is non-trivial. On one hand, it is impossible to train a single model to cover an infinite number of domains. On the other hand, training individual models for each domain requires lots of annotated samples, which induces significant data collection and labeling costs.

In this paper, we tackle the practical yet challenging test-time domain adaptation (TTA) problem, where the model learns from labeled source domain data and adapts with the unlabeled target domain data during test-time. Compared with conventional unsupervised domain adaptation (UDA) (Long et al., 2015), TTA does not need access to source data during adaptation. This setting avoids the privacy issue of sharing the source data and is desirable for real-world applications.

The key challenges of TTA lie in two folds. First, how to effectively modulate the source domain initialized model given noisy unsupervised learning objective. Tent (Wang et al., 2020) optimizes the parameters of batch normalization layers, which is parameter-efficient but lacks adaptation capacity. On the other hand, SHOT (Liang et al., 2020) tunes the feature encoder; AdaContrast (Chen et al., 2022) trains the whole model. Given the current over-parameterized models, these methods are prone to overfitting to the unreliable unsupervised learning objective, especially when the amount of target domain data is limited. Second, given only unlabeled target domain data, use what kind of learning objective for optimization. Existing works propose to use unsupervised objectives, including entropy minimization (Wang et al., 2020), self-supervised auxiliary task (Sun et al., 2019b), pseudo labeling (Lee et al., 2013), or a combination of above objectives (Liang et al., 2020; Chen et al., 2022; Liu et al., 2021b). However, these unsupervised objectives are either not well aligned with the main task (Liu et al., 2021b), or give noisy supervision (Liang et al., 2020).

In this work, we propose **D**ata-**e**fficient test-time adaptation via **P**rompt **T**uning (DePT). Our recipe is simple yet effective, where two key ingredients are proposed to address the two challenges mentioned above, see in Fig. 2. First, a collection of learnable visual prompts (Jia et al., 2022) are trained with labeled source data alongside the vision Transformer (Dosovitskiy et al., 2020). In the adaptation phase, we only finetune the prompts and the classification head, while freezing the backbone parameters. Although the knowledge learned in the source domain is retained in the unchanged backbone, tuning solely the prompts can effectively modulate the model for target domain adaptation. Second, for the learning objective given only unlabeled target domain data, DePT bootstraps the source domain initialized model by a memory bank refined pseudo labeling mechanism. To further alleviate the accumulation of errors during self-training, we designed a hierarchical self-supervised regularization term for the prompts to encourage better target domain representation learning. These two objectives are complementary and jointly optimized.

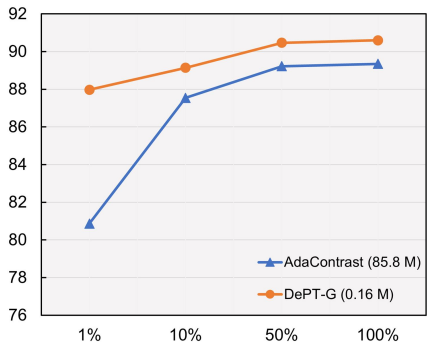

Figure 1: The test-time adaptation performance of different methods with respect to the data ratio on the VisDA dataset. The number in the legend denotes the number of tunable parameters. DePT-G outperforms previous SOTA AdaContrast on 100% target data with only 0.19% tunable parameters. The superiority of DePT is more significant on the low data settings.

The two ingredients in DePT offer multiple merits and lead to state-of-the-art performance on domain adaptation benchmarks: VisDA-C (Peng et al., 2017) and DomainNet-126 (Peng et al., 2019). Tuning prompts during test-time is much more parameter-efficient than full fine-tuning. As illustrated in Fig. 1, with only 0.19% tunable parameters, DePT achieves 1.2% higher performance than previous state-of-the-art AdaContrast (Chen et al., 2022). Moreover, parameter efficiency brings data efficiency to DePT. Compared with AdaContrast, DePT has significant improvement in the low data regime. For example, with only 1% of unlabeled target domain data in VisDA-C, DePT achieves 87.97% average accuracy, which surpasses AdaContrast by 7.1%. DePT also demonstrates superior performance with 85.9% average accuracy in VisDA-C under the more challenging online TTA setting, where test samples come sequentially in a stream. Besides, we also show that DePT is flexible to extend to more DA scenarios like multi-source (Peng et al., 2019) domain adaptation.

2 Related Work

2.1 Domain Adaptation

Domain adaptation (DA) aims at reducing the performance drop introduced by distribution shift. Some highly-related DA settings are introduced in the following.

Unsupervised domain adaptation (UDA) setting is commonly used, where unlabeled data from the testing distribution (target domain) is available during training along with the labeled data from the training distribution (source domain). Many studies attempt to solve this problem using style transfer (Hoffman et al., 2018; Tang et al., 2021; Taigman et al., 2016), feature alignment (Long et al., 2015; Sun et al., 2017; Peng et al., 2019) or learning domain-invariant feature via adversarial training (Ganin et al., 2016; Tzeng et al., 2017). However, the UDA setting has a strong assumption that the source and target domain data are both accessible during training, which is not always true as it is difficult to access source data after the model is deployed. Moreover, these methods usually need to retrain the whole framework if new domains come.

Test-time adaptation (TTA)/Source-free domain adaptation (SFDA) is a more challenging and practical setting where no source domain data is available during adaptation. Such property makes TTA avoid the privacy issue and transmission burden of source data sharing, and enables the pipeline of training the model once and adapting the model to any unknown test distributions. Some recent works explore the TTA setting with different unsupervised objectives and model modulation methods. Test-time training (TTT) (Sun et al., 2019b) introduces a self-supervised rotation prediction task as the test-time optimization objective. Tent (Wang et al., 2020) modulates the batch normalization

parameters by minimizing entropy. SHOT (Liang et al., 2020) adapt the feature extractor to the target domain with a combination of entropy minimization and pseudo labeling. AdaContrast (Chen et al., 2022) tunes the whole model with memory bank-based pseudo labeling and contrastive loss. DePT has several critical differences from these approaches. Most of these methods are proposed for ConvNet, while we switch to vision transformer. For model modulation, we freeze the backbone parameters and learn the target-specific prompts. For the learning objective, we jointly optimize an online memory bank pseudo labeling loss and a hierarchical self-supervised loss specially designed for the prompts. DePT is also flexible on a variety of DA scenarios.

2.2 Self-supervised Learning

Self-supervised learning methods have made great progress in unsupervised visual representation learning. Early attempts use pretext tasks, e.g. colorization (Zhang et al., 2016), rotation prediction (Gidaris et al., 2018), jigsaw (Noroozi & Favaro, 2016), as the self-supervised objective. Another large body of works focus on discriminative approaches (Chen et al., 2020; He et al., 2020), named contrastive learning, which push the representation of positive pairs closer and spread negative pairs further. Recent works show that with a carefully designed structure or learning objective, unsupervised features can be learned without discrimination between instances (Grill et al., 2020; Caron et al., 2021). Besides unsupervised visual representation learning, researchers also explore applying self-supervised objectives for model robustness (Hendrycks et al., 2019) and domain adaptation (Sun et al., 2019a;b). In our method, inspired by BYOL (Grill et al., 2020) and DINO (Caron et al., 2021), we propose a hierarchical self-supervised regularization for the visual prompts that are jointly optimized with pseudo labels.

2.3 Prompt Tuning

The concept of prompt originates from NLP. Prompts are text instructions that are prepended to the input as a hint to inform the pre-trained language model about the task (Brown et al., 2020). Besides text prompts that are mostly constructed via heuristics, soft prompts, i.e., continuous vectors that are optimized via gradient descent, achieve comparable performance with full fine-tuning but have much less tunable parameters (Li & Liang, 2021; Liu et al., 2021a). Prompt tuning is also explored in vision tasks. L2P (Wang et al., 2022) is proposed for continual learning. VPT (Jia et al., 2022) is a highly related work proposed for transfer learning. VPT inserts randomly initialized prompts to the pre-trained model, then optimizes them with the downstream task's label. In contrast, we initialize the prompts with the source domain data and optimize them with unlabeled target domain data.

3 Method

We address the close-set test-time domain adaptation (TTA) setting where source data is not accessible during adaptation. In source domain training, n_s labeled samples $\{x_s^i, y_s^i\}_{i=1}^{n_s}$ from the source domain $\mathcal{D}_s$ are given, where $x_s^i \in \mathcal{X}_s$ and $y_s^i \in \mathcal{Y}_s$ are data and the corresponding labels. A model has a general architecture with parameters θ trained to learn the function $f_s : \mathcal{X}_s \to \mathcal{Y}_s$. We assume the model is parameterized by a neural network and consists of two modules: the feature encoder g_s: $\mathcal{X}_s \to \mathbb{R}^d$ and a classifier h_s: $\mathbb{R}^d \to \mathbb{R}^C$, where d is the dimension of the feature and C is the number of classes. During target domain adaptation, unlabeled data $\{x_t^i\}_{i=1}^{n_t}$ from target domain $\mathcal{D}_t$ is available, where $x_t^i \in \mathcal{X}_t$. The goal of TTA is to adapt the source domain trained model f_s to target domain $f_t : \mathcal{X}_t \to \mathcal{Y}_t$. We assume the target domain share the same label space with source, i.e., $\mathcal{Y}_s = \mathcal{Y}_t = \mathcal{Y}$.

3.1 Visual Prompt Tuning

How to modulate the model during TTA is a non-trivial problem. The optimization objective of TTA is unsupervised and usually noisy, making full parameter tuning sensitive, easy to diverge, and prone to overfitting to the noise, especially when the number of target data is limited. On the other hand, tuning little parameters has limited adaptation capacity. See Appendix B.1 for examples. Therefore, we propose to adopt visual prompt tuning to solve this dilemma.

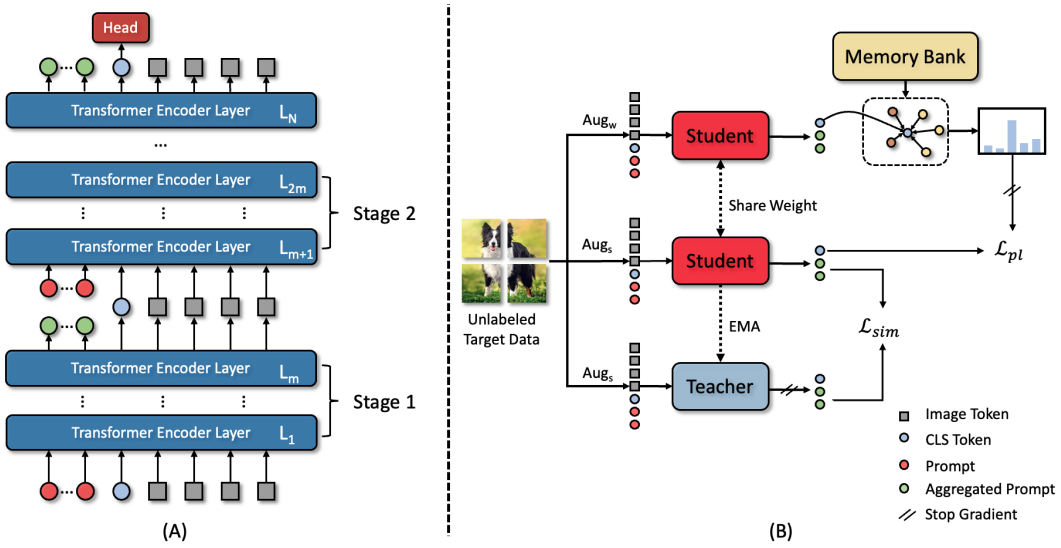

Figure 2: The overview of DePT. We plug prompts into ViT, as shown in (A). We split ViT into multiple stages and prepend prompts to the input of each stage. The prompts along with the backbone parameters are initialized with labeled source domain data. During adaptation, only prompts and the classification head (in red) are finetuned, while the backbone is frozen (in blue). By varying the number of stages and prompts, we can easily alter the number of tunable parameters for different adaptation capacities. (B) The proposed online memory bank refined pseudo-labeling and hierarchical self-supervised objective for test-time training.

Visual prompts are a set of learnable continuous parameters that are prepended to the input of Transformer layers, see in Fig. 2 (A). For a Vision Transformer (ViT), the input image is first divided into patches and then embedded into d-dimensional image tokens $\boldsymbol{E}_0$. Similar to Jia et al. (2022), we concatenate the prompts with CLS token and image tokens to the input of Transformer layers after the image embedding layer. In order to adjust the adaptation capacity, DePT inserts new prompts at different levels of Transformer layers. To be specific, we split the Transformer layers into M stages, where each stage has $m = N/M$ layers. A collection of p prompts $\boldsymbol{P}$ are inserted into the first layer of each stage by concatenating with the image and CLS tokens. We select $M = \{1, 4, 12\}$ to form three variants of DePT: DePT-Shallow (S), DePT-Group (G) and DePT-Deep (D). The prompt augmented ViT is formulated as:

$$
\begin{aligned}
[\boldsymbol{c}_i, \boldsymbol{Z}_i, \boldsymbol{E}_i] &= L_i([\boldsymbol{c}_{i-1}, \boldsymbol{P}_{i-1}, \boldsymbol{E}_{i-1}]) \qquad i = 1, m+1, 2m+1, \ldots, N-m \\
[\boldsymbol{c}_i, \boldsymbol{Z}_i, \boldsymbol{E}_i] &= L_i([\boldsymbol{c}_{i-1}, \boldsymbol{Z}_{i-1}, \boldsymbol{E}_{i-1}]) \qquad i = else \\
\boldsymbol{y} &= head(\boldsymbol{c}_N)
\end{aligned} \tag{1}
$$

where $\boldsymbol{c}_i$ and $\boldsymbol{Z}_i$ represent the CLS token and the aggregated prompt features computed by the i-th Transformer layer, respectively. By altering the number of stages M and the number of prompts p, we can easily control the number of tunable parameters for different adaptation capacity.

The prompt augmented ViT model learns the function $f_s : \mathcal{X}_s \rightarrow \mathcal{Y}_s$ by labeled source data with cross-entropy loss, where all parameters, including the prompts and the backbone, are optimized. In the test-time adaptation phase, we initialize the target model with the source model's weights. We only fine-tune the parameters of the prompts and the classification heads, while all other parameters are fixed. The knowledge learned in the source domain is retained in the frozen backbone. The prompts are finetuned to learn the target-specific knowledge to adjust the representations nonlinearly through self-attention. The classification head is trained to learn a new decision boundary.

3.2 Learning Objective

Our learning objective consists of two parts. First, an online pseudo labeling with memory bank refinement mechanism is used to bootstrap the source domain learned representation to the target

domain. Second, we design a self-supervised objective on the CLS token and the aggregated prompt to forma hierarchical regularization to further reduce the noise in the pseudo labels and improve the target representation quality.

3.2.1 PSEUDO LABELING

Pseudo labeling has widely been used in semi/self-supervised learning. Inspired by Tarvainen & Valpola (2017); Sohn et al. (2020); Chen et al. (2022), we use a student-teacher model and an online memory bank refinement to generate pseudo labels. To be specific, see in Fig. 2 (B), the student f_t and the teacher f_t' share the same architecture, and are both initialized with the source model's weight θ_s at the beginning of adaptation. The teacher is not updated by gradients, but by exponential moving average (EMA) from the student's weights. The teacher maintains an memory bank that stores the feature embeddings and predicted probabilities of target samples. The memory bank is online updated as a first-in-first-out queue from training mini-batches. Given an unlabeled target domain data x_t, a weak and a strong random image transformation: $\mathcal{T}_w$ and $\mathcal{T}_s$ are applied. The embedding of the weakly augmented view is first obtained from the student. The pseudo label is then obtained by a soft voting, i.e. average of the probability, from the top-k nearest neighbors in the memory bank: $\hat{y}_t = \arg\max \mathcal{V}(f_t(\mathcal{T}_w(x_t)))$, where $\mathcal{V}$ denotes the soft voting with memory bank. The student is trained to enforcing the prediction for the strong-augmented view to match $\hat{y}_t$ with a cross-entropy loss:

$$\mathcal{L}_{pl} = \frac{1}{n_t} \sum_{i=1}^{n_t} H(\hat{y}_t, f_t(\mathcal{T}_s(x_t))), \tag{2}$$

where $H(a, b) = -a \log b$. The details can be found in Appendix A.1. Intuitively, the student is fast updating during training, making the pseudo label produced by the student to be noisy. The EMA updated teacher can be seen as an ensemble of the student along training. Thus the memory bank maintained by the teacher stores more stable and high-quality predictions. The soft voting further leverages nearby data point cluster to produce more accurate pseudo labels.

3.2.2 HIERARCHICAL SELF-SUPERVISED REGULARIZATION

Due to the domain shift, the model trained on the source domain cannot directly produce good representations of the target data, i.e., the representation of images from the same class may spread apart. Although using the student-teacher model and the memory bank refinement can significantly improve the quality of pseudo-labels, they still cannot completely avoid the accumulation of errors during self-training. Therefore, we design a hierarchical self-supervised regularization for the proposed prompt augmented ViT to improve the model's representation of target data.

For the prompt augmented ViT, the CLS token of the output of ViT encoder $\boldsymbol{c}_N$ serves as the holistic image representation for classification. The prompts work similarly to the CLS token: prepending to the input of Transformer layers. With self-attention, the prompts can inject domain-specific information into other tokens, and aggregate instance features from other tokens. Each prompt will attend to different attributes and features of the image. The prompts at the output of the Transformer layer, which we call aggregated prompts, serve as a fine-grained representation of the image. Moreover, the aggregated prompts from different stages may attend to distinctive levels of features and thus form a hierarchical representation. Inspired by Caron et al. (2021) and Grill et al. (2020), given different views of the same image, we push both their holistic feature representations (CLS token) and their hierarchical fine-grained representations (aggregated prompts) closer.

We use the DINO (Caron et al., 2021) framework here with a few key modifications. The student encoder g_t and EMA teacher encoder g_t' are reused with the ones in the pseudo labeling section. A series of projection heads q_t: $\mathbb{R}^d \rightarrow \mathbb{R}^K$ are introduced to project the CLS token $\boldsymbol{c}_t$ or the aggregated prompts $\boldsymbol{Z}_i$ of the target sample x_t into K dimensions, where each stage of the prompt augmented ViT has its own projection head. Two random strong-augmented views of the target sample is generated. One is passed to the student, and the other one is passed to the teacher. The projected CLS token and aggregated prompts are obtained from the student: $\tilde{\boldsymbol{c}}_t = q_t(\boldsymbol{c}_t)$, $\{\tilde{\boldsymbol{Z}}_i = q_t(\boldsymbol{Z}_i)\}_{i \in \mathcal{I}}$ and similarly for the teacher $\tilde{\boldsymbol{c}}_t'$, $\{\tilde{\boldsymbol{Z}}_i^{'}\}_{i \in \mathcal{I}}$, where $\mathcal{I} = \{m, 2m, \ldots, N\}$ is the index of the last Transformer layer in each stage. For the holistic regularization on the CLS token, we apply the DINO loss. The probability distribution of the student output is obtained by applying a softmax

function: $P_t(x_t) = \mathrm{softmax}(\tilde{\boldsymbol{c}}_t)$, and so is the teacher's $P'_t(x_t)$. The student aims at matching the teacher's predicted distribution by minimizing the cross-entropy loss:

$$\mathcal{L}_{ss_cls} = H(P'_t(x_t), P_t(x_t)). \tag{3}$$

The technique of centering and sharpening of DINO are also used to avoid collapse, where the details can be found in Appendix A.2. For the hierarchical fine-grained regularization, we minimize the mean squared error between the aggregated prompts of the student and teacher:

$$\mathcal{L}_{ss_prompt} = \frac{1}{M} \sum_{i \in \mathcal{I}} \sum_{j=1}^{p} \left(2 - 2 \cdot \frac{\langle \tilde{\boldsymbol{Z}}_i^j, \tilde{\boldsymbol{Z}}_i^{j\prime} \rangle}{||\tilde{\boldsymbol{Z}}_i^j||_2 \cdot ||\tilde{\boldsymbol{Z}}_i^{j'}||_2} \right). \tag{4}$$

In addition, although the regularization is applied on the aggregated prompts one by one, i.e. the i-th aggregated prompt of the student is pushed closer to the i-th one of the teacher, we find the model tend to learn the trivial solution where all aggregated prompt become similar. To avoid this, we add a diversity term to encourage different prompts to attend to different features, thereby enabling them to learn rich target-specific knowledge. Specifically, we maximize the cosine distance among the p aggregated prompts of the student:

$$\mathcal{L}_{ss_div} = \frac{1}{M} \sum_{i \in \mathcal{I}} \sum_{j=1}^{p} \sum_{k=j}^{p} \left(1 - \frac{\langle \tilde{\boldsymbol{Z}}_i^j, \tilde{\boldsymbol{Z}}_i^k \rangle}{||\tilde{\boldsymbol{Z}}_i^j||_2 \cdot ||\tilde{\boldsymbol{Z}}_i^k||_2} \right), \tag{5}$$

where j and k are the index of the aggregated prompts at the output of the i-th Transformer layer. Therefore, DePT minimizes the following loss function for TTA:

$$\mathcal{L} = \alpha \mathcal{L}_{pl} + \beta_1 \mathcal{L}_{ss_cls} + \beta_2 \mathcal{L}_{ss_prompt} - \lambda \mathcal{L}_{ss_div}. \tag{6}$$

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate the effectiveness and versatility of DePT in a variety of TTA scenarios, covering two major domain adaptation benchmarks below.

VisDA-C Peng et al. (2017) is a large-scale domain adaptation benchmark that focuses on 12-class synthesis-to-real object recognition task. We use the training set as the source and the validation set as the target. The source domain contains 152k synthetis images generated by rendering 3D models while the target domain has 55k real object images sampled from Microsoft COCO.

DomainNet-126 Peng et al. (2019) provides images from multiple domains for classification task. Since the original DomainNet dataset has noisy labels, we follow the previous work (Saito et al., 2019; Chen et al., 2022) to use a subset of DomainNet, which contains 126 classes from 4 domains (Real, Sketch, Clipart, Painting). Besides one-on-one adaptation, we also explore multi-source adaptation on the DomainNet-126 dataset.

Models Different from previous works that use ConvNet as the backbone, e.g. ResNet26/50/101, we use ViT-B as a strong backbone network for adaptation. We use the model and ImageNet pre-trained weight from the timm library (Wightman, 2019) in our experiments.

Baselines We compare our method with both classical UDA methods and test-time adaptation methods. For UDA methods, we compare to DANN (Ganin & Lempitsky, 2015), CDAN (Long et al., 2018), CAN (Kang et al., 2019), SWD (Lee et al., 2019) and MCC (Jin et al., 2020). Note that all UDA methods need access to both source and target domain data, while ours doesn't. For TTA methods, we compare with Tent (Wang et al., 2020), SHOT (Liang et al., 2020), and AdaContrast (Chen et al., 2022).

4.2 VISDA-C DATASET RESULTS

DePT demonstrates state-of-the art performance with much less tunable parameters. Tab. 1 compares DePT with UDA and TTA methods. Without access to source data, DePT with more strong ViT-B backbone achieves better performance than the UDA methods. Among TTA methods,

Table 1: Classification accuracy (%) on VisDA-C train → val adaptation. Upper table: UDA methods. Middle table: TTA methods. Bottom table: online setting. All models with ViT-B backbone are reproduced by us, while the results with R101 backbone are copied from the original paper. Bold and underline denote the first and second high results.

Method	Backbone	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Avg.
DANN	R101	81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
CDAN	R101	85.2	66.9	83.0	50.8	84.2	74.9	88.1	74.5	83.4	76.0	81.9	38.0	73.9
CAN	R101	97.0	87.2	82.5	74.3	97.8	96.2	90.8	80.7	96.6	96.3	87.5	59.9	87.2
SWD	R101	90.8	82.5	81.7	70.5	91.7	69.5	86.3	77.5	87.4	63.6	85.6	29.2	76.4
MCC	R101	88.7	80.3	80.5	71.5	90.1	93.2	85.0	71.6	89.4	73.8	85.0	36.9	78.8
Source	R101	57.2	11.1	42.4	66.9	55.0	4.4	81.1	27.3	57.9	29.4	86.7	5.8	43.8
Tent	R101	86.5	23.0	79.8	51.9	78.3	17.6	87.6	64.2	79.9	23.6	63.9	1.1	57.3
SHOT	R101	95.3	87.5	78.7	55.6	94.1	94.2	81.4	80.0	91.8	90.7	86.5	**59.8**	83.0
AdaCon	R101	97.0	84.7	84.0	77.3	96.7	93.8	91.9	**84.8**	94.3	93.1	94.1	49.7	87.2
Source	ViT-B	99.0	69.1	78.2	71.9	87.7	58.4	96.4	35.0	52.7	92.5	96.0	17.6	71.2
AdaCon	ViT-B	**99.5**	**94.2**	91.2	83.7	98.9	97.7	**96.8**	71.5	96.0	**98.7**	**97.9**	45.0	89.2
DePT-S (Ours)	ViT-B	98.7	90.1	87.4	70.7	98.9	96.0	**96.8**	75.2	91.9	97.9	96.2	52.5	87.7
DePT-G (Ours)	ViT-B	99.2	93.0	93.4	85.6	**99.4**	**98.6**	96.7	76.2	98.2	97.9	96.6	52.0	90.6
DePT-D (Ours)	ViT-B	99.4	93.8	**94.4**	**87.5**	**99.4**	98.0	96.7	74.3	**98.4**	98.5	96.6	51.0	**90.7**
Online Setting														
AdaCon	ViT-B	95.4	75.2	83.4	**64.2**	95.8	93.2	92.4	65.3	92.6	82.9	**95.1**	39.7	81.3
DePT-G (Ours)	ViT-B	**98.3**	**87.6**	**88.9**	62.1	**98.4**	**95.2**	**95.9**	**67.1**	**95.2**	**97.3**	94.9	**50.1**	**85.9**

we reimplement the state-of-the-art AdaContrast (Chen et al., 2022) with the ViT-B backbone for a fair comparison. Even with much less tunable parameters, DePT-G (0.16M) and DePT-D (0.47M) outperforms the strong baseline AdaContrast (85.8M) by $+1.4\%$ and $+1.5\%$. For DePT-S, with only 0.048M tunable parameters, it still has a huge improvement ($+16.5\%$) compared with no adaptation. Although the backbone parameters of ViT are frozen, tuning the prompt at the input can effectively adjust the representation for the target domain.

DePT gains more advantages under limited data settings. Previous TTA methods usually assume all target domain data (testing data) can be used for adaptation. However, although no label is required, obtaining a large amount of target domain data still requires high costs in some real-world applications. It is highly desirable that adaptation methods can adapt with as little target data as possible. Therefore, we investigate the impact of the target domain data quantity on the performance of the TTA methods by evaluating the TTA performance with 1%, 10%, 50%, and 100% of target data, see in Fig. 3 (A). DePT shows strong data efficiency. DePT-G still has 87.97% average accuracy under 1% data, which is only 2.59% lower than with 100% data. In contrast, AdaContrast's average accuracy is only 80.87% at 1% data, which is 8.48% lower than 100% data. For DePT-D, its adaptation capacity is the highest when target data is abundant, e.g., in 100% data. DePT-D has more decline than DePT-G and DePT-S under 1% data, but is still superior to AdaContrast. For DePT-S, the absolute performance is not high, but it shows the best robustness against data amount changes, which only decreases 1.17% under 1% data compared to 100% data.

DePT has strong performance on the more challenging online TTA setting. In the online setting, the model adapts to target data that comes in batch sequentially in a stream. Each data batch can only be used for training once. We do not apply learning rate decay and turn off the memory bank and directly use the model prediction as the pseudo label. The bottom of Tab. 1 shows the performance of DePT online. DePT achieves 85.9% average accuracy, surpassing AdaContrast by +4.6%.

4.3 DomainNet dataset results

DePT consistently improves across seven domain shifts. Tab. 2 shows the comparison between DePT and other TTA methods on the seven domain shifts in the DomainNet-126 dataset. All three variants of DePT show consistent improvement on all seven domain shifts compared with no adaptation. DePT-G performs the best among them and achieves 81.7% average accuracy. Although the proposed DePT-G and DePT-D have much smaller tunable parameters, they outperform the previous SOTA AdaContrast. In the online setting, DePT achieves 3.1% higher performance than AdaContrast. We also find that models with ViT-B backbone demonstrate stronger robustness against domain shifts than the R50 backbone.

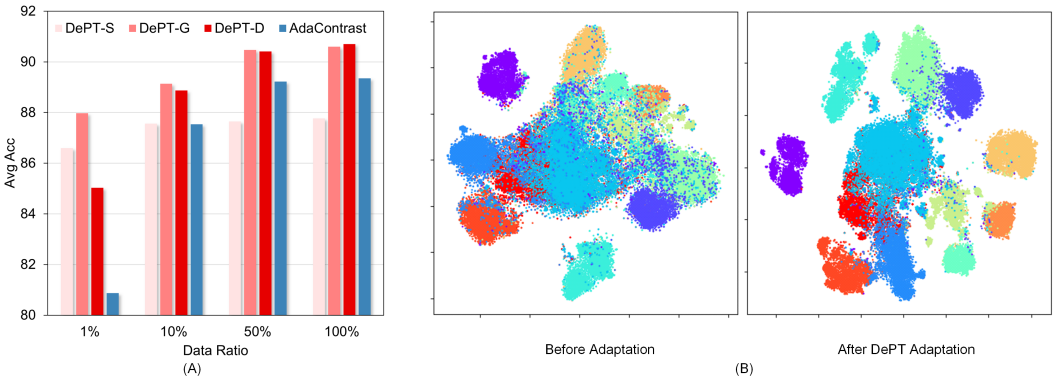

Figure 3: (A) Performance comparison on VisDA-C dataset with different data ratio for TTA. DePT shows less performance degradation when the training data reduces. (B) t-SNE visualization of the DePT-G model before and after adaptation on VisDA-C dataset. Different color denotes different classes. By only tuning the prompts, DePT can effectively adapt the model to the target domain.

Table 2: Classification accuracy (%) of seven domain shifts on DomainNet-126. All models with ViT-B backbone are reproduced by us, while the performance of other baselines with R50 backbone are copied from the original paper. Bold and underline denote the first and second high results.

Method	**Backbone**	**R→C**	**R→P**	**P→C**	**C→S**	**S→P**	**R→S**	**P→R**	**Avg.**
Source	R50	55.5	62.7	53.0	46.9	50.1	46.3	75.0	55.6
TENT	R50	58.5	65.7	57.9	48.5	52.4	54.0	67.0	57.7
SHOT	R50	67.7	68.4	66.9	60.1	66.1	59.9	80.8	67.1
AdaContrast	R50	70.2	69.8	68.6	58.0	65.9	61.5	80.5	67.8
Source	ViT-B	68.7	75.9	69.8	67.7	74.1	60.4	86.4	71.8
AdaContrast	ViT-B	81.8	<u>81.8</u>	82.0	75.8	**82.2**	73.7	**90.4**	81.1
DePT-S (Ours)	ViT-B	81.2	81.4	80.9	75.1	<u>81.7</u>	73.1	88.9	80.3
DePT-G (Ours)	ViT-B	**83.3**	81.6	<u>82.6</u>	**77.8**	81.3	**75.1**	<u>89.9</u>	**81.7**
DePT-D (Ours)	ViT-B	<u>82.8</u>	**82.5**	**82.8**	<u>76.9</u>	81.2	<u>74.6</u>	89.8	<u>81.5</u>
Online Setting									
AdaContrast	ViT-B	72.1	77.9	71.8	70.1	75.7	65.5	**87.9**	74.2
DePT-G (Ours)	ViT-B	**74.8**	**78.3**	**74.2**	**72.5**	**77.2**	**68.5**	87.2	**76.1**

Beyond one-on-one adaptation. We further evaluate DePT to multi-source TTA setting, details of the setting can be found in Appendix A.3. See in Tab. 3, DePT makes the best use of multi-source domain knowledge by utilizing a set of domain-specific prompts and a shared domain-agnostic prompt. In contrast, SHOT (Liang et al., 2020) makes one-by-one domain adaptation separately and then makes predictions by an ensemble, which does not jointly optimize for multi-source domains.

4.4 Analysis

Visual prompt tuning helps to learn better target representation. Fig. 3 (B) shows the t-sne visualization of the target feature embedding before and after adaptation. Before the adaptation, the feature embeddings of the source model on the target data are mixed and difficult to distinguish due to domain shift. After adaptation, the embeddings are more scattered, and the number of samples that fall into the wrong cluster is less. This experiment proves that although the backbone is fixed, DePT still learns the knowledge of the target domain by the prompt and improves the target representation through attention layers.

Prompt tuning is highly effective for adaptation. Tab. 4 shows the impact of the number of prompts on VisDA-C dataset. Increasing the number of stages or the number of prompts will introduce more tunable parameters during TTA. As the VisDA-C validation set has abundant unlabeled

Table 3: Classification accuracy (%) of multi-source and multi-target DA setting on DomainNet-126 dataset. $\mathcal{R}$ denotes the **remaining** three domains except the single source.

Multi-source ($\mathcal{R} \rightarrow$)	**Backbone**	$\mathcal{R} \rightarrow$**R**	$\mathcal{R} \rightarrow$**C**	$\mathcal{R} \rightarrow$**S**	$\mathcal{R} \rightarrow$**P**	**Avg.**
ERM	ViT-B	88.7	76.7	69.2	78.7	78.3
SHOT	ViT-B	90.7	83.1	75.9	83.4	83.2
DePT-G (Ours)	ViT-B	**91.0**	**83.7**	**78.3**	**84.0**	**84.2**

Table 4: Ablation study of the effect of number of prompts in VisDA-C dataset. We provide the average TTA accuracy and the number of tunable parameters.

Prompt Num	**1**		**10**		**50**		**100**	
Model	Acc	Params/M	Acc	Params/M	Acc	Params/M	Acc	Params/M
DePT-S	83.8	0.009	86.0	0.016	87.7	0.048	87.6	0.086
DePT-G	85.4	0.010	88.4	0.040	90.6	0.16	90.6	0.32
DePT-D	86.3	0.018	89.0	0.10	90.7	0.47	90.4	0.94

data, increasing tunable parameters usually results in better adaptation performance, e.g., DePT-G-100 and DePT-D-50 achieve SOTA adaptation performance. Surprisingly, even with only one prompt, DePT still achieves decent adaptation accuracy.

Table 5: Ablation of the contribution of each component to the final performance on VisDA-C.

#	**PL**	**MB**	**CLS Reg.**	**Prompt Reg.**	**Prompt Div.**	**DePT-G**
0						73.2
1	✓					85.6
2	✓	✓				86.9
3	✓	✓	✓			89.3
4	✓	✓	✓	✓		88.2
5	✓	✓	✓	✓	✓	90.6

Contribution of each loss. Tab. 5 shows the contribution of each loss to the final performance. The pseudo labeling and memory bank refinement can make better pseudo label generation, and achieves the performance of 86.9. By adding the DINO regularization on the CLS token, the model learns better representation. This regularization works jointly with pseudo labeling to the performance of 88.3. We tried to further add hierarchical regularization on the prompts. However, direct adding regularization makes prompts to collapse. We further introduce the diversity loss of prompts and finally boost the performance to 90.6.

5 Conclusion

In this paper, we propose a data-efficient test-time adaptation approach via visual prompt tuning. We pretrain a collection of visual prompts alongside the network with labeled source domain data. In the adaptation phase, we only adapt the prompts to learn target-specific knowledge with the backbone fixed. A flexible scheme is proposed to insert a certain number of prompts to the input of each stage of the Transformer. By doing so, we can easily control the adaptation capacity and tunable parameters by altering the number of stages and prompts. An online pseudo labeling with memory bank refinement is jointly learnt with a hierarchical self-supervised objective for test-time prompt tuning. We extensively evaluate the proposed method with VisDA-C and DomainNet-126 datasets and show that our simple recipe can lead to SOTA performance in multiple settings and superior data efficiency. In addition, we further verify the effectiveness and flexibility of DePT on online and multi-source adaptation settings.

REFERENCES

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.

Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 295–305, 2022.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pp. 1180–1189. PMLR, 2015.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

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

Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. *Advances in neural information processing systems*, 32, 2019.

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pp. 1989–1998. Pmlr, 2018.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. *arXiv preprint arXiv:2203.12119*, 2022.

Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. Minimum class confusion for versatile domain adaptation. In *European Conference on Computer Vision*, pp. 464–480. Springer, 2020.

Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4893–4902, 2019.

Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10285–10295, 2019.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, pp. 896, 2013.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

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*, pp. 6028–6039. PMLR, 2020.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021a.

Yuejiang Liu, Parth Kothari, Bastien van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. Ttt++: When does self-supervised test-time training fail or thrive? *Advances in Neural Information Processing Systems*, 34:21808–21820, 2021b.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105. PMLR, 2015.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pp. 69–84. Springer, 2016.

Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge, 2017.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1406–1415, 2019.

Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8050–8058, 2019.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

Baochen Sun, Jiashi Feng, and Kate Saenko. Correlation alignment for unsupervised domain adaptation. In *Domain Adaptation in Computer Vision Applications*, pp. 153–171. Springer, 2017.

Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825*, 2019a.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A Efros, and Moritz Hardt. Test-time training for out-of-distribution generalization. 2019b.

Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016.

Zhiqiang Tang, Yunhe Gao, Yi Zhu, Zhi Zhang, Mu Li, and Dimitris N Metaxas. Crossnorm and selfnorm for generalization under distribution shifts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 52–61, 2021.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7167–7176, 2017.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

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*, pp. 139–149, 2022.

Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pp. 649–666. Springer, 2016.

A IMPLEMENTATION DETAILS

A.1 PSEUDO LABELING DETAILS

During adaptation, we use a student-teacher model and an online memory bank refinement to generate pseudo labels. The student and the teacher share the same prompt-augmented ViT architecture. They are both initialized with the source model's weight θ_s at the beginning of adaptation. As the student is fast updating during training, the pseudo label directly produced by the student is usually unstable and noisy. We introduce a slow momentum teacher model whose weights θ'_texponential moving average updated by the student's weights θ_t:

$$\theta'_t = m\theta'_t + (1-m)\theta_t, \tag{7}$$

where θ_t is the student's weights, θ'_t is the teacher's weights, and m is the momentum for updating. The teacher can be seen as an ensemble of student along training, thus is more stable and has higher pseudo label accuracy.

The teacher maintains an online memory bank B with size N to refine the pseudo labels from the student via soft voting, where the memory bank stores the feature embeddings and the predicted probabilities $\{\boldsymbol{c}'_i, \boldsymbol{p}'_i\}_{i=1}^{N}$ of weakly augmented target data from the teacher. The memory bank B is initialized by randomly selected N samples, and then updated with data from training mini-batch as a first-in-first-out queue.

Given an unlabeled target domain data x_t, a weak and a strong random image transformation $\mathcal{T}_w$ and $\mathcal{T}_s$ are applied. The student produces the feature embedding c_t based on the weakly augmented view $\mathcal{T}_w(x_t)$. To refine the prediction of the student, we use c_t to retrieve top-k nearest neighbors in the memory bank B by measuring the cosine distance between $\boldsymbol{c}_t$ and $\{\boldsymbol{c}'_i\}_{i=1}^{N}$. The refined hard pseudo label is obtained by averaging the probability of the top-k neighbors followed by a $\arg\max$ operation:

$$\hat{y}_t = \arg\max \frac{1}{k}\sum_{i=1}^{k} p'_i \tag{8}$$

Inspired by FixMatch (Sohn et al., 2020), the student is trained to enforce the cross-entropy loss against the model's output for the strongly-augmented view:

$$\mathcal{L}_{PL} = \frac{1}{n_t}\sum_{i=1}^{n_t} H(\hat{y}_t, f_t(\mathcal{T}_s(x_t))) \tag{9}$$

A.2 SELF-SUPERVISED OBJECTIVE DETAILS

We use the DINO (Caron et al., 2021) framework for self-supervised learning with a few key modifications. We reuse the student and teacher model in the pseudo labeling with the DINO framework, which are both initialized by the source model. We apply the DINO loss on the CLS token. The sharpening and centering operation is also used to avoid collapse. To be specific, the sharpening operation is applied for both the student and teacher:

$$P_t(x_t)^{(i)} = \frac{\exp(\tilde{\boldsymbol{c}}_t^{(i)}/\tau)}{\sum_{k=1}^{K}\exp(\tilde{\boldsymbol{c}}_t^{(k)}/\tau)} \tag{10}$$

The teacher is further applied a centering operation:

$$P'_t(x_t)^{(i)} = \frac{\exp((\tilde{\boldsymbol{c}}_t'^{(i)} - \boldsymbol{c}^{(i)})/\tau')}{\sum_{k=1}^{K}\exp((\tilde{\boldsymbol{c}}_t'^{(k)} - \boldsymbol{c}^{(k)})/\tau')} \tag{11}$$

where τ and τ' are the temperature for the student and teacher, respectively. $\boldsymbol{c}$ is the center that is exponential moving average updated for more stable estimation. The multi-crop and local-to-global regularization are not applied in DePT. More details can be found the (Caron et al., 2021).

A.3 EXTENDING TO MULTI-SOURCE DOMAIN TTA SETTING

The multi-source domain test-time adaptation setting aims at training a model from multiple source domains and adapt it to the unseen target domain. Many works aims at learning a domain-invariant feature across source domains. However, such methods are different to learn domain-invariant features if the source domain is diverse. They also can't capture domain-specific knowledge.

Our method is versatile to extend to multi-source domain test-time adaptation setting. See in Fig. 4, the basic idea is that all domains share a domain-agnostic backbone while each domain has its own prompts to learn the domain-specific knowledge. During source training, each domain prompts along with a shared general prompts are randomly initialized. In each training iteration, we attractively sample images from each domain. The backbone, shared prompts and the corresponding domain prompts are optimized via supervised loss. During the TTA phase, prompts from all domain are averaged and concatenated with the shared prompts as the initialization. Then the optimization is the same as one-on-one TTA adaptation.

In our experiments, we use three domains in the DomainNet-126 as the source domains, and use the left one as the target.

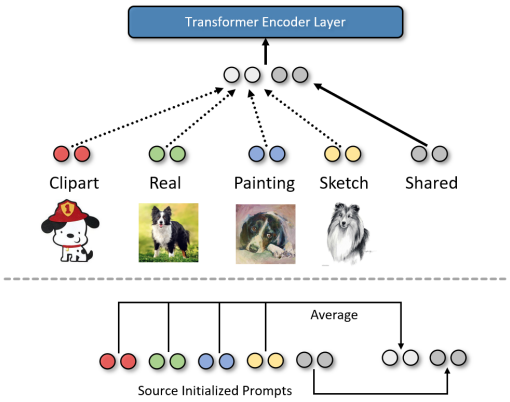

Figure 4: Our method is easy to extend to multi-source/multi-target domain adaptation setting, where only the corresponding phase needs to be modified.

A.4 IMPLEMENTATION DETAILS

We use PyTorch for all implementation. For source training, we initialized the ViT-B-16 with ImageNet pretrained weights from timm library (Wightman, 2019). We follow Chen et al. (2022) to randomly split the source data with 90% for training and 10% for validation. For source training, we train around 11k iterations. For target training, we train around 6k iterations for all datasets unless noted. For all experiments, we use SGD optimizer with momentum 0.9 and weight decay 1e-4, and cosine annealing on the learning rate. The training batch size is set as 128.

B MORE EXPERIMENT RESULTS

B.1 MOTIVATION OF USING PROMPT TUNING

How to modulate the model is vital for the TTA setting. Since there is only unlabeled target data during adaptation, the model can only learn through the unsupervised learning objective. Current unsupervised learning objectives are not always true and usually exist some noises and errors. If there are too many parameters to modulate, the current over-parameterized model is prone to over-fitting to the noise of the unsupervised objective, especially when the amount of unlabeled target data is limited. On the other hand, if modulating too few parameters, although overfitting will be avoided, the adaptation capacity of the model will be limited.

Here we perform an experiment to verify our claim. We use AdaContrast (Chen et al., 2022) and Tent (Wang et al., 2020) as examples to modulate the model. Among them, AdaContrast finetunes all parameters of the model. Tent only adjusts the transformation parameters of the BatchNormalization layers, which only contain very few parameters. For ViT, we adjust the transformation parameters of the LayerNorm (LN) layer instead. For these two model modulation approaches, we use the learning objective of AdaContrast for training, i.e., MoCo-based self-supervised combined with pseudo labeling.

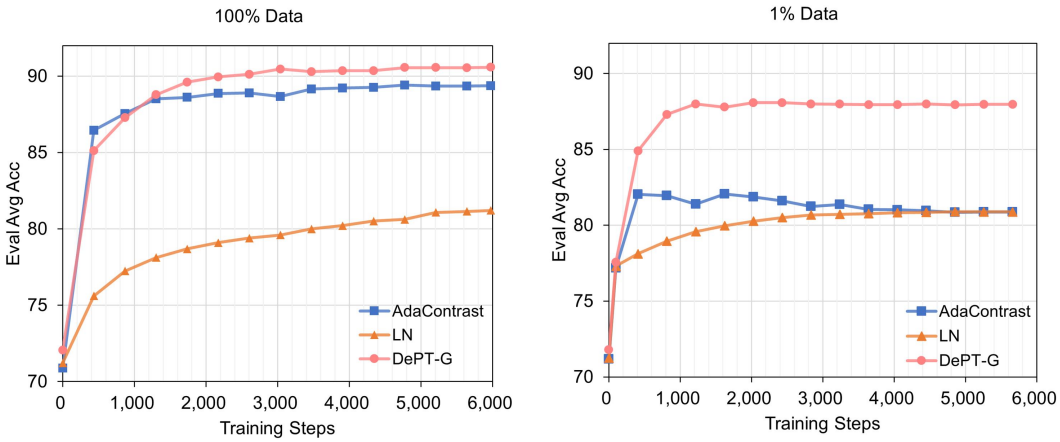

Figure 5: The curve of average accuracy of evaluation v.s. training steps with 100% data and 1% data on VisDA-C dataset.

Fig. 5 shows the results of training the above two modulation approaches with 100% and 1% VisDA-C unlabeled target domain data. Under 100% data, AdaContrast adapts well to the target domain. This is because the target domain of VisDA-C has 55K sufficient data, making AdaContrast not easy to overfit. For LN, its performance is limited by the adaptation capacity. Under 1% of the data, with the increasing of training steps, AdaContrast over-fits obviously, as the performance of adaptation gradually declines. LN does not overfit but is still limited by the adaptation capacity, and the performance is not very good. In the setting of TTA, no target domain label is available, and we currently do not have a metric to know when to stop adaptation.

A method that can avoid over-fitting and have enough adaptation capacity is necessary. Therefore, we are motivated to explore using visual prompt tuning to solve this dilemma.

B.2 ABLATION ON THE HYPER-PARAMETER CHOICE

Tab. 6 shows the impact of the hyper-parameter choice, including learning rate, the teacher temperature in the DINO loss, and the weight to balance training loss.

For visual prompt tuning, we find that a larger learning rate would benefit the performance. For example, learning rates of 1e-3 and 5e-3 lead to a decent performance. Continuing to amplify the learning rate may lead to instability in training.

The Teacher temperature is a hyper-parameter in the DINO loss that controls the sharpness of the teacher output distribution. A large temperature makes the teacher tend to output a uniform distribution, which makes the initial optimization difficult. The performance of small temperature is slightly worse. Using a warmup mechanism on the teacher's temperature is a good balance.

We finally investigate the effect of weights among the proposed losses in Eq. 6 . We set α to 1, and adjust the value of $\beta_1 = \beta_2 = \beta$. We set $\lambda = \beta/20$. We found that a small β value results in better performance, e.g., below 0.1.

Despite the relative performance varies by different hyper-parameter choices, DePT still consistently demonstrates state-of-the-art performance compared to other methods, which also shows the robustness of DePT against the choice of hyper-parameters.

Table 6: Ablation of the impact of hyper-parameter choices of DePT-G on VisDA-C.

LR	2e-4	5e-4	1e-3	5e-3
Avg Acc	89.6	90.0	90.4	90.6
Teacher Temp	0.04	0.05	0.07	0.04-0.07
Avg Acc	90.1	90.2	90.6	90.5
β	0.01	0.05	0.1	0.5
Avg Acc	90.1	90.2	90.6	89.8