STYLIP: Multi-Scale Style-Conditioned Prompt Learning for CLIP-based Domain Generalization

Shirsha Bose1* Ankit Jha2* Enrico Fini3 Mainak Singha2* Elisa Ricci3 Biplab Banerjee2
1Technical University of Munich, Germany 2Indian Institute of Technology Bombay, India 3University of Trento, Italy

shirshabosecs@gmail.com, ankitjhal6@gmail.com, enrico.fini@unitn.it
mainaksingha.iitb@gmail.com, e.ricci@unitn.it, getbiplab@gmail.com

Abstract

Large-scale foundation models, such as CLIP, have demonstrated impressive zero-shot generalization performance on downstream tasks, leveraging well-designed language prompts. However, these prompt learning techniques often struggle with domain shift, limiting their generalization capabilities. In our study, we tackle this issue by proposing STYLIP, a novel approach for Domain Generalization (DG) that enhances CLIP’s classification performance across domains. Our method focuses on a domain-agnostic prompt learning strategy, aiming to disentangle the visual style and content information embedded in CLIP’s pre-trained vision encoder, enabling effortless adaptation to novel domains during inference. To achieve this, we introduce a set of style projectors that directly learn the domain-specific prompt tokens from the extracted multi-scale style features. These generated prompt embeddings are subsequently combined with the multi-scale visual content features learned by a content projector. The projectors are trained in a contrastive manner, utilizing CLIP’s fixed vision and text backbones. Through extensive experiments conducted in five different DG settings on multiple benchmark datasets, we consistently demonstrate that STYLIP outperforms the current state-of-the-art (SOTA) methods.

1. Introduction

Advancements in large-scale vision and language models, such as CLIP [50] and ALIGN [18], have made remarkable progress in computer vision tasks. These models employ contrastively trained vision and text encoders to capture semantically meaningful concepts in a shared embedding space. They demonstrate impressive zero-shot generalization performance using text prompts like A photo of a [CLS]. However, designing an optimal prompt is challenging, and recent studies focus on data-driven prompt optimization [69]. Despite their success, prompt learning is limited to the training data distribution and is susceptible to domain shift [9]. Domain shift, common in real-world applications, poses challenges as deep learning models are sensitive to differences between training and test data distributions [17]. To tackle this, researchers explore Domain Generalization (DG) [28, 67, 71], which aims to learn a domain-agnostic representation from multiple datasets sourced from different domains for application to novel target domains. Traditional DG techniques rely on vision encoders trained exclusively on image data [70,72,73]. Recent efforts combine foundation models with prompt engineering [50, 69] to bridge the semantic gap, but their practical applicability in DG settings requires further exploration.

In this paper, we focus on a more challenging setting where significant visual variations exist across different domains, unlike existing prompting methods [38, 68, 69, 74] that evaluate CLIP’s generalization capabilities on datasets with limited domain shift (e.g., variants of ImageNet [24]). Fig. 1 illustrates the average multi-source DG performance
on DomainNet [48], where zero-shot CLIP [50] underperforms compared to the best traditional DG model, SWAD [6], by approximately 2.5%. Using a domain-conditional prompt (A [Domain] of a [CLS]) boosts CLIP’s accuracy by nearly 3%, highlighting the importance of a representative prompt for DG. However, domain-level annotations are not always available, and the static domain name may not capture the style properties characterizing the domains [42].

Existing specialized prompt-tuning techniques [64, 68, 74] improve CLIP’s performance (Fig. 1), but their effectiveness for DG is uncertain as prompts refined from random vectors may not effectively encode domain knowledge [68, 74]. Zhang et al. [64] propose a domain-prompt initialization strategy based on batch statistics of visual features but overlook important lower-level style characteristics and consider domain-level supervision. Another recent approach [45] learns prompts from CLIP without utilizing visual samples from the source domains but incorporates textual domain knowledge.

These discussions highlight a research gap in learning prompts that account for unknown domain shifts without explicit domain identifiers. We argue that leveraging visual features in such scenarios is crucial, along with dynamically incorporating object-level variations into the prompts to aid in cross-domain generalization tasks [68]. Motivated by these considerations, our research question is whether we can utilize CLIP’s vision backbone to encode image style and content information for learning domain and instance-aware prompts to address DG.

Our proposed STYLIP: We introduce STYLIP, a novel generic prompt tuning strategy for CLIP that addresses these challenges. Our approach aims to enhance the prompts’ understanding of class concepts by conditioning them on domain and content information derived from CLIP’s visual space. To achieve this, we leverage the hypothesis that instance-wise feature statistics from intermediate layers of an image encoder capture the visual domain information [35]. We extract mean and standard deviation values from CLIP’s intermediate feature map outputs and utilize a set of STYLE PROJECTORS to learn domain-specific tokens in the prompts. Unlike existing models such as [68, 69, 74] that learn prompt token embeddings from ad-hoc sentences, our approach benefits from using style features at different scales, which leads to improved domain-aware prompts and better prompt initialization.

Additionally, we propose to incorporate image content information into the prompt embeddings to capture object-level variations and avoid overfitting to the training classes, which is particularly important in DG scenarios concerning disjoint training and test classes. While Zhou et al. [68] addresses this issue by adding high-level semantic features from CLIP’s vision encoder to prompt tokens, we consider a DG setting where the distributions of training and test classes differ [40]. To achieve this, we combine visual feature responses from different layers of CLIP’s vision encoder and aggregate them through a CONTENT PROJECTOR. By encoding mid- to high-level image characteristics that are more generic across categories [66], we aim to enhance transferability across domains/classes. Unlike the literature [22, 68], we propose a learnable fusion network to aggregate these visual features with the final prompt embeddings obtained previously.

Contributions: We highlight our major contributions as:

- We introduce STYLIP, a domain-unified prompt learning strategy that leverages CLIP’s frozen vision encoder to extract the domain and content information from an image and deploy them in prompt learning through light-weight learnable style and content projectors.

- We acquire prompt tokens from visual style features at various scales, facilitating the consolidation of hierarchical domain knowledge, thereby assisting in generalization across different domains. Moreover, we incorporate multi-scale visual content information into prompt embeddings, effectively mitigating overfitting and promoting generalization across different categories.

- We showcase the performance of STYLIP for multiple datasets on five major DG tasks: i) single-source and multi-source DG, ii) cross-dataset DG, iii) in-domain base to novel class DG, and iv) cross-domain base to novel class DG, a novel task we introduce in the context of prompting. Experimentally, STYLIP outperforms the competitors in all tasks at least by 0.2–4%. To our knowledge, ours is the first attempt to extensively study the DG problem using CLIP.

2. Related Works

Domain generalization. The DG problem has different variations. Single-source DG [49, 60] trains with one domain, while multi-source DG [11, 21, 72] considers training multiple domains simultaneously. In a closed-world setting, where the label set is shared across domains, DG approaches commonly address domain shift. Heterogeneous DG [36, 58] faces additional challenges due to different labels between the source and target domains. Previous research on DG proposed methods such as domain alignment losses [20, 32, 33, 59], self-supervised learning [5], ensemble learning [62], domain-specific networks [41], and meta-learning [47]. However, these methods often require more training domains, which can influence DG performance. To overcome this, novel pseudo-domains have been generated using domain augmentation approaches [21, 34, 72, 73]. In single-source DG models [49, 60, 65], diverse styles can be synthesized by perturbing the source domain through entropy maximization, meta-learning, and adversarial learning. Conversely, methods for heterogeneous DG [31, 37, 72] aim to improve model generalizability for novel tasks.

DPL [64] used CLIP [50] for multi-source DG by in-
3. Problem and notation

The DG problem involves \( N \) labelled source domains \( S^i = \{ x^k_i, y^k_i \}_{k=1}^{n_i} \approx P_{\text{data}}^i, 1 \leq i \leq N \), where \( x^i \in X^i \), \( y^i \in Y \), and \( P_{\text{data}}^i \) denote the input data, label, and the joint distribution concerning the data and the label space, respectively. Furthermore, \( P_{\text{data}}^i \neq P_{\text{data}}^j \) for \( i, j \in \{ 1, 2, \cdots, N \} \), indicating that the source domains are mutually distinct. We call the setting single-source DG if \( N = 1 \), else it is known as multi-source DG. The goal is to train a model \( f : X \rightarrow Y \) given \( S = \{ S^i \}_{i=1}^{N} \), which is expected to generalize for a novel target domain \( S^{N+1} = \{ x^k_{N+1}, y^k_{N+1} \}_{k=1}^{n_{N+1}} \) unseen during training with \( x^i \in X^i \) and \( y^i \in Y^i \) and \( P_{\text{data}}^i \) denotes the target distribution which is different from the source distributions. Typically, we consider a closed-set setting where \( Y \cup Y^i = Y \cap Y^i \). Also, for the base to new class generalization setting, we consider \( Y \cap Y^i = \emptyset \).

3.2. The StylIP model

Here, we introduce StylIP, a novel approach for DG based on CLIP [50]. StylIP leverages CLIP’s frozen vision encoder \( f_v \) and text encoder \( f_t \), trained on a large volume of image-text pairs (see Fig. 2). \( f_v \) that transforms an input image into a feature embedding vector can be implemented with different architectures: in our experiments (see Section 4), we consider ResNet-50 (RN50) [13], and ViT-B/16 [12]. \( f_t \) is built upon a Transformer [56]: it is provided with an input of a sequence of word tokens and converts them into a vectorized representation.

As stated, StylIP seeks to utilize the multi-scale visual features extracted from different levels of \( f_v \) to estimate the style and content primitives and further channel them in learning a generic prompt space regarding a concept. Typically, high-level representations of the deepest layer of a vision encoder tend to capture the abstract object semantics suitable for classification but suffer from a lack of description of local patterns like oriented edges or local shapes [66]. Therefore, the set of characteristics obtained from multiple levels is deemed more transferable between tasks than the high-level features alone. Similarly, the instance-wise feature statistics calculated from multiple layers of the encoder capture different levels of style, e.g., the texture in the top layers usually has larger granularity than those in the bottom layers [1].

To model a continuous prompt embedding space using these multilevel visual features, StylIP (see Fig. 2) adopts a set of projector networks on top of \( f_v \) and \( f_t \): a set of \( M \) style projectors \( \{ P_m \}_{m=1}^{M} \) to encode domain characteristics into \( M \) prefix tokens \( \{ c_m \}_{m=1}^{M} \), a content projector \( P_C \) to encode feature responses from all the \( L \) encoder layers of \( f_v \) after reducing their dimensions using bottleneck layers \( \{ B_l \}_{l=1}^{L} \), and a fusion projector \( P_A \). We discuss the structure of the proposed projectors in detail below.

**Embedding multi-level style information into prompt tokens:** For calculating the style features, let us consider the vector \( \mathbf{f}_i(x) = [ \mu_i(x); \sigma_i(x) ] \) denoting the channel-wise mean and standard deviation of the feature map outputs from the \( l \)th layer \( (1 \leq l \leq L) \) of \( f_v \) also indicated as \( f^l_v(x) \). Here, \([\cdot; \cdot]\) denotes the concatenation operation. Specifically, if \( f^l_v(x) \) is of dimensions \( W \times H \times C \) (height, width, and depth dimensions), the statistics corresponding to the \( c \)th feature map \( f^c_v \), \((\mu^c_v, \sigma^c_v)\), are calculated as:

\[
\mu^c_v(x) = \frac{1}{W \times H} \sum_{i,j} f^c_v(x_{ij}), \quad \sigma^c_v(x) = \sqrt{\frac{1}{W \times H} \sum_{i,j} (f^c_v(x_{ij}) - \mu^c_v(x))^2}
\]
Figure 2. STYLIP generates a joint embedding space for image and prompt embeddings, leveraging style and content information extracted from the vision encoder \( f_v \). The projectors \( \{P_m\}_{m=1}^{M}, \{B_l\}_{l=1}^{L} \), and \( P_C \) adapt these features for the target task. For each image \( x \), the style features from layer \( m \) of \( f_v \) are used to learn the \( m \)th token of the prompt, \( c_m(x) \), through the style projector \( P_m \). Similarly, the multi-scale content features are aggregated using \( P_C \) after dimensionality reduction by the bottleneck networks \( B_l \). The fusion projector \( P_A \) computes the classification weights. Unlike existing prompt tuning techniques (CoOp, CoCoOp, and DPL), STYLIP differs in its approach. \( v_m \) represents the prompt learner from a random vector, and \( X_k^t \) denotes a batch of samples for domain \( d \).

\[
\mu^t_l = \frac{1}{W H} \sum_{w,h=1}^{W,H} f_v^c(x)_{w,h} \tag{1}
\]

\[
\sigma^t_l = \sqrt{\frac{1}{W H} \sum_{w,h=1}^{W,H} (f_v^c(x)_{w,h} - \mu^t_l)^2} \tag{2}
\]

In the simplest case, when the context length \( M \) equals the number of encoder layers \( L \), we seek to learn the \( m \)th context vector \( c_m \) from \( F_m(x) \). Considering that the dimensions of \( F_m(x) \)s are inconsistent and to appropriately input \( F_m \) into the text encoder \( f_t \), we deploy the style projectors \( \{P_m\}_{m=1}^{M} \) and compute \( c_m(x) = P_m(F_m(x)) \), i.e. the \( m \)th context vector for the text prompt. We define

\[
t_y = [c_1(x)]\ldots[c_M(x)]|CLS_y| \tag{3}
\]

as the prompt for \((x, y)\) where \([CLS_y] \) is the word embedding of label \( y \). Finally, \( f_t \) generates the embedding \( \hat{f}_t(x) \).

However, the context length in prompting is a hyperparameter, meaning an \( M \) different from \( L \) may be preferred for a given task. To incorporate this flexibility in our prompt learning, we consider aggregation or replication of representations from \( \{F_l\}_{l=1}^{L} \) depending on whether \( M < L \) or \( M > L \), respectively.

**Supplementing the prompt embeddings with multi-scale image content features:** The paradigm of \( t_y \) considers the information of the visual style of the images, but a static class embedding \([CLS]y\) for all images with the label \( y \) may limit its versatility. To further generalize the prompt embeddings, we propose to supplement \( t_y \) with content image information. As discussed, we extract multiple complementary visual characteristics associated with an image by aggregating the multilevel feature responses obtained from the \( L \) blocks of \( f_v \).

One naive way to combine this multilevel information is by flattening the feature maps of individual blocks, followed by concatenation. However, this leads to a very high-dimensional vector representation compared to the dimensionality of \( t_y \), undermining the effects of \( t_y \) in the final classification weights. As a result, the contrastive task may lead to triviality. We propose reducing the feature maps’ dimensions before concatenation. This also shrinks the size of the inputs to \( P_C \), thus controlling its number of learnable parameters and the amount of information exchanged by the two encoders.

Precisely, given the \( l \)th-layer feature maps \( f_v^l(x) \), we perform \( 1 \times 1 \) convolution followed by flattening using \( B_l \) to reduce the channel depth of \( f_v^l(x) \) from original \( C \) to \( \tilde{C} \ll C \), resulting in \( B_l(f_v^l(x)) \in \mathbb{R}^{WH\tilde{C} \times 1} \). Finally, we concatenate the \( B_l(f_v^l(x)) \)s to obtain \( \tilde{f}_v(x) \):

\[
\tilde{f}_v(x) = [B_1(f_v^1(x));B_2(f_v^2(x));\ldots;B_L(f_v^L(x))] \tag{4}
\]

The content projector \( P_C \) learns the combined image embedding \( P_C(\tilde{f}_v(x)) \) through a linear transformation. To generate the classification weights for a given \((x, y)\), we first concatenate \( P_C(\tilde{f}_v(x)) \) with \( f_t(x, y) \) and transform the aggregated information through the fusion projector \( P_A \) to obtain \( \hat{t}(x, y) \) as follows:

\[
\hat{t}(x, y) = P_A(\{P_C(\tilde{f}_v(x));f_t(x, y)\}) \tag{5}
\]

### 3.3. Training and inference

The projectors are trained using a contrastive loss \( \mathcal{L}\_con \) between \( \hat{t}(x, y) \) and the image features obtained from the final embedding layer of \( f_v \), i.e., \( f_v(x) \), as follows:

\[
\mathcal{L}\_con = \arg\min_{\{P_m\}_{m=1}^{M}, \{B_l\}_{l=1}^{L}, P_C, P_A} \mathbb{E}_{(x,y) \sim P^S_{\text{data}}} \log(p(\hat{t}(x,y)|x)) \tag{6}
\]

where \( P^S_{\text{data}} \) is the joint data distribution of \( S \) and

\[
p(\hat{t}(x^k,y^k)|x^k) = \frac{e^{\delta(\hat{t}(x^k,y^k), f_v(x^k))/\tau}}{\sum_{n \in Y} e^{\delta(\hat{t}(x^k,y^k), f_v(x^k))/\tau}} \tag{7}
\]
\(\delta\) defines the cosine similarity and \(\tau\) is the temperature hyperparameter. The contrastive loss synergistically maximizes the similarity between the image and the correct class prompt embeddings while minimizing the similarity between the image and all the opposing classes.

During inference, we calculate the compatibility between \(f_i(x_t)\) and the prompt embeddings for all classes in \(\mathcal{Y}^t\). The class with the highest compatibility is selected as:

\[
\hat{y}_t = \arg\max_{n \in \mathcal{Y}^t} p(i(x_t), n) \| x_t \) \] (8)

### 4. Experimental Results

**Datasets:** We evaluate STYLIP over five benchmark datasets for multi-source and single-source domain generalization, namely Office-Home [57], PACS [29], VLCS [30], Digits-DG [72], and DomainNet [48]. We further analyze the performance of STYLIP for cross-dataset generalization, where STYLIP is trained on ImageNet [24] and tested on ten other different datasets [69]. Detailed descriptions of the datasets are provided in SUPPLE.

**Implementation, training, and evaluation protocols:** We implement the projectors in \((\mathcal{P}_C, \mathcal{P}_A, \{\mathcal{P}_m\}_{m=1}^M)\) as single dense layers. We train the model with Adam optimizer [23] with a learning rate of \(2e - 2\) and betas \((0.9, 0.999)\). We consider a context length of four for all the experiments following [68, 69]. For RN50, we consider the feature map outputs from the four convolution stages to extract the style and content features, hence \(L = M = 4\). For ViT-B/16, we obtain the embedding outputs from the \(L = 12\) encoder layers. We further average the features for every three consecutive layers of \(f_v\) in a bottom-up manner without overlap to generate four intermediate feature representations, which are subsequently used to produce four distinct domain information vectors to be passed to \(\{\mathcal{P}_m\}_{m=1}^4\). As we show in Fig. 3, the feature statistics capture the domain information in ViT-based prompt learning for STYLIP, similar to RN50. We fix the number of output channels of the bottleneck \(C = 3\) using cross-validation, where 10% images from each source domain are treated as the validation set. We further ablate \(C\) in Section 4.2 to check other architecture choices. Finally, we consider a mini-batch size of 4 for DomainNet and Office-Home, while it is 8 for the other datasets, and we train the model for 10 epochs. We report the average top-1 classification performance on \(SN^{+1}\) over three different executions. In terms of model complexity, STYLIP is extremely light-weight and consists of 0.18% more parameters than CoOp and CoCoOp and 0.06% more parameters than MaPLe.

**Baselines:** We consider three types of methods for comparison to check the generalizability of pre-trained CLIP features and that of the prompting strategies. Our baseline is zero-shot CLIP with the prompt as ‘A [Domain] of a [CLS]’. We also include domain name in the prompt as ‘A [Domain] of a [CLS]’. We use CLIP features to train a linear classifier, which we term Linear Probing. Furthermore, we deploy these features in conjunction with the benchmark DG technique of CROSSGRAD [52], where we put the learnable networks on top of frozen CLIP for back-propagation training. From the traditional DG literature, we report the performance of SWAD [6] for Multi-DG and SagNet [44] and DSBF [63] for Single-DG, respectively. Furthermore, we choose to compare STYLIP with existing prompt learning techniques including CoOp [69], CoCoOp [68], CLIP-Adapter [10], DPL [64], ProGrad [74], VPT [19], CSVPT [26], Maple [22] and, TPT [53], etc.

Finally, we evaluate three variants of STYLIP intending to ablate the individual components of our approach: i) \(\{\mathcal{P}_m\}_{m=1}^M\) are trained from random vectors (similar to CoOp), but we consider the multi-scale content feature learning of STYLIP through \(((\{\mathcal{B}_i\}_{i=1}^L, \mathcal{P}_A, \mathcal{P}_C)\), respectively. (STYLIP-con). This establishes the importance of including the visual style information in the prompt tokens. ii) the model without content features and \((\{\mathcal{B}_i\}_{i=1}^L, \mathcal{P}_C)\) but with \(\{\mathcal{P}_m\}_{m=1}^M\) (STYLIP-sty). This is to verify the importance of the multi-scale content features, and iii) the version of STYLIP where the features of the deepest layer of \(f_v\) are used for the content branch together with \(B_L\) only (STYLIP+). This is to assess the importance of the multi-scale content features over the single-scale high-level visual content properties as used in the literature [68].

### 4.1. Comparison with state-of-the-art

We discuss the experimental comparisons of STYLIP with the literature in the following order of the DG tasks:

- **i)** multi-source DG,
- **ii)** single-source DG,
- **iii)** cross-domain base to novel class DG,
- **iv)** in-domain base to novel class DG and,
- **v)** cross-dataset DG, respectively. We follow the leave-one-domain-out evaluation protocol for multi-source DG where all the domains except one are considered source domains while the model is to be verified on the held-out target domain. For single-source DG, we train the model on for a large number of epochs such as VPT [19] trains for 100 epochs, whereas STYLIP is trained only with 10 epochs.
Table 1. Comparison of our proposed STYLiP with the state-of-the-art methods on PACS, VLCS, Digits-DG, Office-Home, and DomainNet datasets for multi-source DG in terms of mean leave-one-out performance. † uses a different backbone than CLIP. In (%)

<table>
<thead>
<tr>
<th>Method</th>
<th>PACS</th>
<th>VLCS</th>
<th>OffHome</th>
<th>Dig-DG</th>
<th>DomainNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWAPD [3]</td>
<td>88.10</td>
<td>79.10</td>
<td>70.60</td>
<td>-</td>
<td>46.50</td>
</tr>
<tr>
<td>Lin. Probing</td>
<td>91.65</td>
<td>79.48</td>
<td>70.17</td>
<td>62.22</td>
<td>46.10</td>
</tr>
<tr>
<td>CROSSGRAD [53]</td>
<td>91.56</td>
<td>79.63</td>
<td>70.47</td>
<td>62.64</td>
<td>46.65</td>
</tr>
<tr>
<td>ZS-CLIP [16]</td>
<td>90.32</td>
<td>76.43</td>
<td>66.75</td>
<td>56.41</td>
<td>44.18</td>
</tr>
<tr>
<td>ZS-CLIP + DN</td>
<td>91.86</td>
<td>-</td>
<td>67.93</td>
<td>-</td>
<td>47.50</td>
</tr>
<tr>
<td>CoCOp [69]</td>
<td>92.28</td>
<td>81.87</td>
<td>71.65</td>
<td>73.11</td>
<td>49.71</td>
</tr>
<tr>
<td>CoCOp [60]</td>
<td>91.64</td>
<td>82.30</td>
<td>71.93</td>
<td>74.58</td>
<td>50.16</td>
</tr>
<tr>
<td>Clip-Adapter [40]</td>
<td>92.08</td>
<td>82.35</td>
<td>72.18</td>
<td>73.79</td>
<td>50.25</td>
</tr>
<tr>
<td>DPL [60]</td>
<td>91.96</td>
<td>82.12</td>
<td>72.34</td>
<td>73.43</td>
<td>50.38</td>
</tr>
<tr>
<td>ProtGrad [74]</td>
<td>92.01</td>
<td>82.23</td>
<td>71.85</td>
<td>74.56</td>
<td>50.27</td>
</tr>
<tr>
<td>TPT [53]</td>
<td>92.16</td>
<td>82.39</td>
<td>72.07</td>
<td>74.68</td>
<td>50.30</td>
</tr>
<tr>
<td>STYLiP-com</td>
<td>92.35</td>
<td>84.07</td>
<td>73.89</td>
<td>75.90</td>
<td>51.43</td>
</tr>
<tr>
<td>STYLiP</td>
<td>92.96</td>
<td>84.39</td>
<td>74.22</td>
<td>76.21</td>
<td>51.80</td>
</tr>
<tr>
<td>STYLiP*</td>
<td>93.47</td>
<td>86.60</td>
<td>73.96</td>
<td>79.31</td>
<td>51.83</td>
</tr>
<tr>
<td>STYLiP</td>
<td>93.99</td>
<td>84.83</td>
<td>74.80</td>
<td>76.49</td>
<td>52.29</td>
</tr>
</tbody>
</table>

Table 2. Comparing STYLiP with CLIP-based SOTA methods for single-source DG on PACS, VLCS, and Office-Home datasets in terms of mean leave-all-but-one-domain-out performance. (In %)

<table>
<thead>
<tr>
<th>Method</th>
<th>PACS</th>
<th>VLCS</th>
<th>OffHome</th>
</tr>
</thead>
<tbody>
<tr>
<td>SagNet [44]</td>
<td>61.90</td>
<td>-</td>
<td>68.00</td>
</tr>
<tr>
<td>DSReF [5]</td>
<td>85.33</td>
<td>-</td>
<td>63.91</td>
</tr>
<tr>
<td>Lin. Probing</td>
<td>85.67</td>
<td>69.42</td>
<td>65.99</td>
</tr>
<tr>
<td>CoG [60]</td>
<td>89.88</td>
<td>74.04</td>
<td>69.04</td>
</tr>
<tr>
<td>CoGoGP [60]</td>
<td>88.69</td>
<td>74.80</td>
<td>69.48</td>
</tr>
<tr>
<td>CLIP-Adapter [10]</td>
<td>88.86</td>
<td>75.31</td>
<td>69.29</td>
</tr>
<tr>
<td>DPL [60]</td>
<td>89.24</td>
<td>74.86</td>
<td>69.10</td>
</tr>
<tr>
<td>ProtGrad [74]</td>
<td>88.55</td>
<td>75.40</td>
<td>69.49</td>
</tr>
<tr>
<td>TPT [53]</td>
<td>88.93</td>
<td>75.02</td>
<td>69.58</td>
</tr>
<tr>
<td>STYLiP-com</td>
<td>90.56</td>
<td>75.92</td>
<td>70.41</td>
</tr>
<tr>
<td>STYLiP</td>
<td>91.77</td>
<td>76.39</td>
<td>70.87</td>
</tr>
<tr>
<td>STYLiP*</td>
<td>89.14</td>
<td>75.67</td>
<td>69.85</td>
</tr>
<tr>
<td>STYLiP</td>
<td>92.61</td>
<td>77.18</td>
<td>71.60</td>
</tr>
</tbody>
</table>

Table 3. Analysis of the generalization from base to new classes across domains. We show results on DomainNet with ClipArt acting as the source domain, while others denote the target. The model is trained (backbone CLIP ViT-B/16) using 16 shots from the base classes. (In %)

<table>
<thead>
<tr>
<th>Method</th>
<th>Base</th>
<th>ClipArt</th>
<th>ClipArt Infograph</th>
<th>Painting</th>
<th>Quick Draw</th>
<th>Real</th>
<th>Sketch</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIP [7]</td>
<td>79.00</td>
<td>76.55</td>
<td>49.80</td>
<td>70.84</td>
<td>17.76</td>
<td>88.11</td>
<td>66.54</td>
</tr>
<tr>
<td>CoG [60]</td>
<td>82.79</td>
<td>73.60</td>
<td>48.80</td>
<td>71.38</td>
<td>20.90</td>
<td>85.19</td>
<td>67.39</td>
</tr>
<tr>
<td>CoGoGP [60]</td>
<td>82.85</td>
<td>77.40</td>
<td>52.61</td>
<td>72.06</td>
<td>20.80</td>
<td>80.00</td>
<td>64.12</td>
</tr>
<tr>
<td>CLIP-Adapter [10]</td>
<td>80.51</td>
<td>78.33</td>
<td>51.70</td>
<td>71.81</td>
<td>20.15</td>
<td>87.70</td>
<td>67.60</td>
</tr>
<tr>
<td>DPL [60]</td>
<td>82.15</td>
<td>78.49</td>
<td>52.20</td>
<td>71.88</td>
<td>20.30</td>
<td>87.74</td>
<td>67.73</td>
</tr>
<tr>
<td>ProtGrad [74]</td>
<td>83.70</td>
<td>77.50</td>
<td>51.44</td>
<td>72.16</td>
<td>20.86</td>
<td>87.11</td>
<td>67.05</td>
</tr>
<tr>
<td>MaPL [22]</td>
<td>84.58</td>
<td>77.41</td>
<td>51.63</td>
<td>71.85</td>
<td>20.51</td>
<td>87.35</td>
<td>67.82</td>
</tr>
<tr>
<td>STYLiP-com</td>
<td>83.71</td>
<td>77.68</td>
<td>52.04</td>
<td>72.53</td>
<td>20.97</td>
<td>87.20</td>
<td>67.70</td>
</tr>
<tr>
<td>STYLiP</td>
<td>84.19</td>
<td>77.62</td>
<td>52.68</td>
<td>75.00</td>
<td>21.10</td>
<td>87.74</td>
<td>68.29</td>
</tr>
<tr>
<td>STYLiP*</td>
<td>83.34</td>
<td>77.30</td>
<td>51.92</td>
<td>72.36</td>
<td>20.95</td>
<td>87.39</td>
<td>67.97</td>
</tr>
<tr>
<td>STYLiP</td>
<td>84.39</td>
<td>78.14</td>
<td>53.09</td>
<td>73.60</td>
<td>21.69</td>
<td>87.90</td>
<td>68.61</td>
</tr>
</tbody>
</table>

**one domain and test it on the remaining domains (leave-all-but-one-domain-out).** We consider the standard few-shot training dataset with 16-shots, following the CLIP literature [22, 69] for all the tasks. However, we have mentioned a detailed sensitivity analysis of STYLiP against the number of available training samples in Fig. 5.

**Discussions on multi-source and single-source DG:** We present the mean leave-out performance of PACS, VLCS, Office-Home, Digits-DG, and DomainNet in Table 1 for both RN50 and ViT-B/16 backbones. Our method, STYLiP, surpasses zero-shot CLIP, Linear Probing, and domain alignment approaches by at least 3% for both backbones, achieving state-of-the-art (SOTA) results. Additionally, STYLiP outperforms competitors, including DPL, CSVPT, and other prompting methods, across all datasets and vision backbones. Notably, when using ViT-B/16, STYLiP achieves outstanding performance with scores of 86.94% on VLCS, 81.38% on Digits-DG, and 62.03% on DomainNet, surpassing others by at least 3%.

Comparatively, the performance of STYLiP-com is slightly lower than STYLiP by approximately 0.5-1.5%, while STYLiP-sty performs marginally better than STYLiP-com but remains inferior to STYLiP. However, both STYLiP-sty and STYLiP-com exhibit comparable or better performance than other prompting methods. The limitations of these variants of STYLiP are that they only capture partial visual properties, leading to sub-optimal prompt learning. In contrast, STYLiP fully leverages both style and content information of images, reducing the gap between visual and semantic spaces. Moreover, STYLiP outputs STYLiP* due to the multi-scale content features, which are more generalizable than deeper semantically oriented visual representations.

In the single-source DG setting, using PACS, VLCS, and Office-Home datasets, we report the average leave-all-but-one-domain-out in Table 2 for all domain combinations. Remarkably, STYLiP achieves a convincing improvement of approximately 1.4-2.5% over other prompting techniques, establishing a new SOTA for single-source DG. For detailed domain-wise results in both single-source and multi-source DG setups, please refer to SUPPLEMENTARY.

**Generalizing across novel domains and categories:** In this experiment on DomainNet, we consider ClipArt as the source domain while the others denote the target domain. We divide the classes equally, and the model is trained and tested on the disjoint class sets, following [40]. In Table 3, STYLiP outperforms the other prompting techniques in nine out of ten cases by $\approx 0.3 - 4\%$ while generalizing to novel classes from both the source and the target
Table 4. Comparison with SOTA methods on base-to-new generalization. STYLIP shows better generalization performance over existing methods on 11 different recognition datasets on 16-shots and a context length of four. HM is the harmonic mean. (In %)

<table>
<thead>
<tr>
<th>Methods</th>
<th>Base</th>
<th>New</th>
<th>HM</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIP [50]</td>
<td>69.34</td>
<td>74.22</td>
<td>71.70</td>
</tr>
<tr>
<td>CoOp [69]</td>
<td>82.69</td>
<td>63.22</td>
<td>71.66</td>
</tr>
<tr>
<td>CoCoOp [68]</td>
<td>82.57</td>
<td>71.69</td>
<td>78.83</td>
</tr>
<tr>
<td>LISP [4]</td>
<td>82.70</td>
<td>74.90</td>
<td>78.61</td>
</tr>
<tr>
<td>MaPLe [22]</td>
<td>82.28</td>
<td>75.14</td>
<td>75.55</td>
</tr>
<tr>
<td>STYLIP-con</td>
<td>82.64</td>
<td>75.39</td>
<td>78.85</td>
</tr>
<tr>
<td>STYLIP- sty</td>
<td>82.93</td>
<td>75.67</td>
<td>79.13</td>
</tr>
<tr>
<td>STYLIP*</td>
<td>82.30</td>
<td>75.24</td>
<td>78.61</td>
</tr>
<tr>
<td>STYLIP</td>
<td>83.22</td>
<td>75.94</td>
<td>79.41</td>
</tr>
</tbody>
</table>

domains, respectively. For the Real domain of DomainNet, STYLIP lags [68] by a mere 0.21%. STYLIP is less prone to overfitting to the classes of the source domain due to the better transferability offered by our model through multi-scale feature embedding. To validate this, we repeat this experiment using the model STYLIP*, which deals with only the deepest layer visual encodings. Confirming our hypothesis, we find that the performance of STYLIP* is consistently poorer than STYLIP for all cases ($\approx 0.2 \sim 1.2\%$).

In-domain base to novel class generalization: In addition to the cross-domain generalization to novel categories, we show the performance of STYLIP on the 11 datasets [69] where the base and novel classes are divided for each dataset to define the source and the target domains. A context length of four and 16 samples per class are considered for training the model. We depict the average performance over all the datasets in Table 4, which shows that STYLIP beats the state-of-the-art, CoOp [69], CoCoOp [68], LISP [4], and MaPLe [22] convincingly by more than 0.8% on average H-score (H-score is the harmonic mean of the base-class and novel-class accuracies). Specifically, STYLIP is better than CoOp and CoCoOp by $\approx 8\%$ and $4\%$, respectively. We observe that STYLIP is able to beat the others both for the base as well as novel classes. This is important since the existing methods are mostly found to boost the performance of novel classes at the cost of decreasing base class performance. Refer to Supplementary for the detailed results.

Figure 4. Effects of using style features from an increasing number of layers from $f_v$ for the fixed content features from the $L^{th}$ layer, and using content features from an increasing number of layers for a fixed style feature ($M = 1$).

Figure 5. Sensitivity of prompting techniques (with ViT-B/16) on the context length ($M$) and the number of training samples per class. We show the average performance for multi-source DG on the PACS dataset.

4.2. Ablation analysis

Generalization across datasets: Following the literature [69], we perform prompt learning using 16-shots from the 1000 classes of ImageNet (source) and test on the other 10 datasets (target). On the source domain, STYLIP beats the recent [22] by almost 1.6% (Table 5). In contrast, for more specialized target datasets, such as DTD, EuroSAT and FG-Vaircrafts, STYLIP beats the other competitors. For the fine-grained datasets, STYLIP shows improvements up to 2%, exhibiting much stronger transferability.

Analysis of multi-scale features: For the PACS and VLCS datasets with RN50, we conducted two experiments to investigate the impact of multi-scale features from $f_v$ on performance. In the first experiment, we focused on the content feature corresponding to the $L^{th}$ layer of $f_v$, varying $M$ from 1 to 4 for the style features. In the second experiment, we fixed $M = 1$ for the deepest layer style features and varied $L$ from 1 to 4 for the content features. As shown in Figure 4, increasing the number of layers for both style and content features positively influenced the performance.

In another experiment, we seek to show the usefulness of the multi-scale content features. In this regard, we compare STYLIP with a multi-scale version of CoCoOp [68] where we combine the multi-scale features to the input tokens instead of the deepest layer features as done in the CoCoOp paper. It can be observed from Tab. 6 that STYLIP is able to beat MS-CoCoOp on multiple datasets for the Multi-DG task. This can be attributed to the improved prompting of STYLIP using the disentangled style and content features.

Context length ($M$): As we mention in Fig. 5, we evaluate the effects of different context lengths for multi-source DG on PACS using the ViT backbone. We find that STYLIP outperforms the other techniques, including [10, 64, 68, 69, 74] for context lengths of 1, 4, 6, 12, and 16. To generate the style primitives for $M = 16$, we choose to replicate the feature statistics vectors for the final four en-
Sensitivity to the number of training samples: tentively across all context lengths. performance of \([68, 74]\), while S

Depth of style and content projectors: shows improvements with more shots.

grades in the low-data regime, while \([10, 64, 74]\) shows com-

LIP performs consistently across all context lengths.

Sensitivity to the number of training samples: To assess the robustness of \(\text{StyLIP}\) versus the number of training samples for the conventional DG setting, we train the single-source DG model on PACS while varying the number of training samples per class in the range \([1, 5, 10, 16, \text{All}]\). As shown in Fig. 5, the DG performance of \([68, 69]\) degrades in the low-data regime, while \([10, 64, 74]\) shows comparatively better performance. Finally, \(\text{StyLIP}\) maintains its superior performance for very few training samples and shows improvements with more shots.

Depth of style and content projectors: To check the sensitivity of \(\text{StyLIP}\) on the depth of \(\{P_m\}_{m=1}^{16}\) and \(P_C\), we consider cases of multi-source DG where the projectors are two-layers and three-layers deep with a consistent number of nodes per layer, respectively, in PACS and Office-Home (Tab. 7). We find that performance decreases marginally with increasing depth: 0.6 – 0.8% for PACS and 0.2 – 0.4% for Office-Home than \(\text{StyLIP}\) with linear projectors, suggesting \(\text{StyLIP}\) is indeed lightweight.

Learnable vs. non-learnable \(P_A\): Typically, \(P_C\) and \(f_i\) produce feature embeddings of similar dimensions; hence, one way to fuse them in \(P_A\) is through element-wise feature pooling. In this regard, we use the max and average feature pooling strategies and observe in Tab. 7 that such aggregations affect the performance, reducing the multi-source DG accuracies on PACS and Office-Home by \(\approx 2 – 3\%\) in max pooling and \(\approx 4 – 5\%\) in average pooling than \(\text{StyLIP}\).

Analysis of style features: Typically, the mean and std. of the feature maps together are known to capture the visual style information. To validate the same, we study the model’s performance with either mean or std. being used as input to the style projectors. In this regard, we see a decrease in the performance of \(1 – 2\%\) compared to \(\text{StyLIP}\), suggesting the importance of both statistical estimates. Interestingly, we see better accuracy when only std. is used for context learning than only mean (Tab. 7).

Sensitivity to the depth of the bottleneck layer \(C\): We consider different \(C\) values in the range 2, 3, 4, 16 to see the effects of the bottleneck dimensions in the final accuracy (Tab. 7). While \(C = 3\) provides the best performance, we see the numbers decreasing from \(C = 4\) onwards, finally producing a dip of almost 1.5% for \(C = 16\). Besides, we consider the scenario where \(1 \times 1\) convolutions are not used, and we perform global average pooling (GAP), or directly flatten the feature maps and then concatenate. Both options perform poorly compared to \(\text{StyLIP}\) by \(0.5 – 1\%\).

5. Takeaways

In this paper, we aim to address the challenge of domain shift in DG tasks by proposing \(\text{StyLIP}\), a domain-agnostic prompt learning strategy for CLIP. By disentangling and incorporating multi-scale visual style and content information from CLIP’s frozen vision encoder into the prompt learning process, we enhance its generalizability. Extensive evaluations on various cross-domain inference tasks demonstrate the consistent state-of-the-art performance of \(\text{StyLIP}\). Our study on task-generalizable prompt learning paves the way for new research opportunities in computer vision. Future directions could explore domain-aware prompt learning with different foundation models and extend the proposed approach to structured prediction tasks.
References


