CLAP: Isolating Content from Style through Contrastive Learning with Augmented Prompts

Yichao Cai, Yuhang Liu, Zhen Zhang, and Javen Qinfeng Shi

Australian Institute for Machine Learning, University of Adelaide, SA 5000, Australia
{yichao.cai,yuhang.liu01,zhen.zhang02,javen.shi}@adelaide.edu.au

Abstract. Contrastive vision-language models, such as CLIP, have garnered considerable attention for various dowmsteam tasks, mainly due to the remarkable ability of the learned features for generalization. However, the features they learned often blend content and style information, which somewhat limits their generalization capabilities under distribution shifts. To address this limitation, we adopt a causal generative perspective for multimodal data and propose contrastive learning with data augmentation to disentangle content features from the original representations. To achieve this, we begin with exploring image augmentation techniques and develop a method to seamlessly integrate them into pretrained CLIP-like models to extract pure content features. Taking a step further, recognizing the inherent semantic richness and logical structure of text data, we explore the use of text augmentation to isolate latent content from style features. This enables CLIP-like model's encoders to concentrate on latent content information, refining the learned representations by pre-trained CLIP-like models. Our extensive experiments across diverse datasets demonstrate significant improvements in zeroshot and few-shot classification tasks, alongside enhanced robustness to various perturbations. These results underscore the effectiveness of our proposed methods in refining vision-language representations and advancing the state-of-the-art in multimodal learning. [1]

Keywords: Data Augmentation · Latent Variables · Disentanglement

1 Introduction

Vision-language models, exemplified by CLIP [36], have garnered substantial attention due to their exceptional generalization capabilities, achieved through the learned features, obtained by utilizing a cross-modal contrastive loss [20, 25, 36]. However, despite being pre-trained on extensive datasets, CLIP-like models fall short in disentangling latent content information and latent style information. Consequently, they are not immune to spurious correlations, i.e., style-related information is erroneously utilized to predict task-related labels. These limitations become evident in the presence of distribution shifts or adversarial attacks where spurious correlations often change across different environments. For examples,

---

[1] Our code is available at `https://github.com/YichaoCai1/CLAP`.

(a) Image augmentation

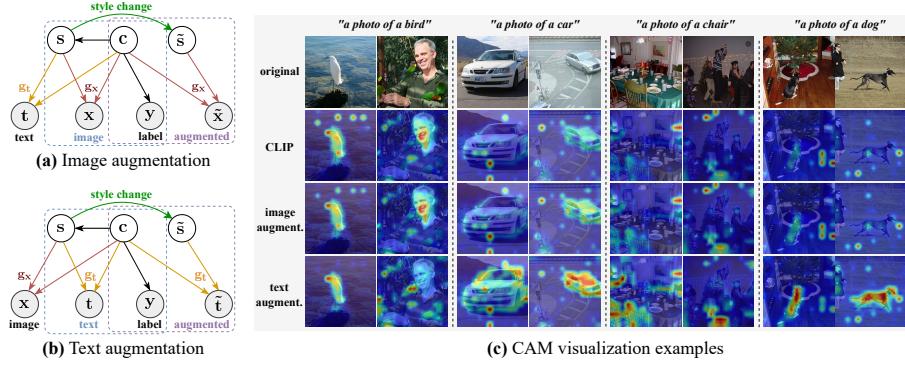**(b)** Text augmentation

(c) CAM visualization examples

Fig. 1: Causal generative models of vision-language data. Image and text data are generated through distinct underlying deterministic processes, $\mathbf{g_x}$ for images and $\mathbf{g_t}$ for texts, derived from a unified latent space with latent content variables $\mathbf{c}$ and latent style variables $\mathbf{s}$. Latent content $\mathbf{c}$ exclusively determines the sample label $\mathbf{y}$. **(a)** Soft interventions on latent style variables $\mathbf{s}$ result in $\tilde{\mathbf{s}}$, subsequently generating augmented images $\tilde{\mathbf{x}}$. **(b)** Due to the same latent space, soft interventions on latent style variables $\mathbf{s}$ can also result in $\tilde{\mathbf{s}}$, producing augmented text $\tilde{\mathbf{t}}$. **(c)** A qualitative comparison of image features for zero-shot classification using "a photo of a [class]" prompts, visualized using class activation map (CAM) [32], demonstrates that while image augmentation can enhance CLIP features, the features obtained through text augmentation methods predominantly focus on the content.

(1) a notable dependence on specific input text prompts has been reported for zero-shot capabilities [21,47,48]; (2) performance decline in few-shot scenarios has been observed in few-shot learning scenarios [13,36]; and (3) susceptibility to adversarial attacks has been explored [33,43,45].

Taking a causal perspective, this work begin with a simple yet effective method, image augmentation, to disentangle content and style information within the learned representations of CLIP-like models. This approach is inspired by recent advancements in theoretical development in causal representation learning [41], which demonstrate that augmented image can be interpreted as a result of soft interventions on latent style variables, as depicted in Fig. 1a. Such augmentation results in a natural data pair where content information remains unchanged while style information changes. Consequently, using contrastive learning, it becomes feasible to isolate the invariant content information from the variant style information. Motivated by this theoretical advancement, we propose a practical method to incorporate image augmentation into CLIP-like models to extract content information from the original learned features. Specifically, a disentangled network is designed to fine-tune the pre-trained CLIP model by using a contrastive loss with image augmentation.

Despite the advancements made in disentangling content and style information from the original features learned by CLIP-like models through image augmentation, we recognize an inherent limitation: it is generally challenging to

design adequate image augmentations to ensure all style factors change in an image. Theoretically, disentangling content and style information necessitates changes in all style factors [41]. However, inducing sufficient changes in latent style through image augmentation poses challenges due to the high dimensionality and complexity of style information in image data. Achieving significant style variation via artificially designed image augmentation techniques, such as transforming a photograph of a dog into a sketch while preserving content but dramatically altering style, is notably difficult.

Taking a further step, rather than relying on image augmentation, we explore the use of text augmentation to disentangle latent content and style factors. This shift is motivated by two key observations: 1) Vision and language data share the same latent space. Therefore, text augmentation can also be utilized to induce changes in latent style factors instead of image augmentation. 2) Text data inherently possesses high semanticity and logical structure, making it more amenable to property-wise manipulation compared to image data. Consequently, implementing sufficient style changes through text augmentation is more feasible than image augmentation, contributing to isolating content from style information, see Fig. 1c for visual comparison. For instance, transforming text from "a photo of a dog" to "a sketch of a dog" is straightforward in the language modality, whereas achieving a similar transformation in image data is challenging. Inspired by these observations, we posit that introducing style variations through text augmentation, as illustrated Fig. 1b, provides a more effective approach for learning vision-language content features than relying on image augmentation.

In summary, our contributions include: (1) Aimed at disentangling latent content and style factors to refine vision-language features of pre-trained CLIP-like models, we propose constrastive learning with data augmentation to fine tune the original features of pre-trained CLIP-like models from a causal perspective. (2) We present a novel method customized for pre-trained CLIP-like models. This method leverages a disentangled network, which is trained using contrastive learning with image augmentation, to extract latent content features from the learned features provided by image encoder of CLIP-like models. (3) We propose Contrastive Learning with Augmented Prompts (CLAP), to extract latent content features from representations of CLIP-like models. It begins by training a disentangled network using the pre-trained text encoder of CLIP-like models and text augmentation. Subsequently, the trained disentangled network is transferred to the image encoder of CLIP-like models. (4) Experiments conducted on a large real dataset demonstrate the effectiveness of the proposed image augmentation and text augmentation in terms of zero-shot and few-shot performance, as well as robustness against perturbations.

2   Related Work

Contrastive Vision-Language Models Using a cross-modal contrastive loss, CLIP [36] revolutionarily introduced a scalable contrastive vision-language model

by leveraging a large corpus of internet-sourced image-text pairs, demonstrating unprecedented zero-shot learning capabilities and exceptional generalization ability across datasets and supporting numerous downstream tasks [38]. ALIGN [20] expanded the scale of contrastive vision-language modeling, training on up to one billion image-text pairs while integrating the vision transformer's self-attention mechanism [11], which further enhanced performance. Despite their successes, CLIP-like models exhibit sensitivity to input text prompts [21, 48], leading to variable performance across different prompts. Efforts to mitigate this prompt sensitivity through prompt learning and engineering [9,14,21,47,48] focus on customizing prompts for specific tasks but do not fundamentally enhance CLIP's representations. Furthermore, CLIP-like models are vulnerable to adversarial attacks [4, 12], with current strategies [33, 45] involving adversarial-natural image pairs to improve resilience. Our work diverges from task-specific approaches by aiming to enhance CLIP's representations from a disentanglement perspective, addressing the aforementioned issues inherent in CLIP-like models.

Disentangled Representation Learning Aimed at segregating intrinsic latent factors in data into distinct, controllable representations, disentangled representation learning benefits various applications [24,40,44]. Specifically, in classification tasks, it's shown that enhancing the model's performance and robustness against data distribution perturbations can be achieved by more effectively disentangling invariant content variables, without needing to identify all intrinsic latent variables completely [22,26–28]. Within single modalities, studies such as [49] have illustrated that contrastive learning [7,16,18] can potentially reverse the data generative process, aiding in the separation of representations. Furthermore, [41] suggest that image augmentation can help isolate content variables from the latent space through significant stylistic changes. [19] employs mixture techniques for data augmentation, enabling more abundant cross-modal matches. Diverging from these methods, our approach focuses on employing text augmentation to disentangle latent content variables, introducing a unique approach to learn refined vision-language representations.

3   A Causal Generative Model for Multi-Modal Data

To understand pretrained CLIP-like models, we investigate the underlying causal generative process for vision-language data. We consider the following causal generative model as depicted in Fig. 1. In the proposed model, the shared latent space ruling vision and language data is divided into two distinct sub-spaces: one corresponding to the latent content variables $\mathbf{c}$ and the other to the latent style variables $\mathbf{s}$. The latent content variables are posited to determine the object label $\mathbf{y}$, a relationship corroborated by prior studies [22,29,31]. Furthermore, to elucidate the correlation between the latent style variable $\mathbf{s}$ and the object variable $\mathbf{y}$, our model incorporates the premise that the latent content variable $\mathbf{c}$ causally influences the latent style variable $\mathbf{s}$, in concordance with the principles of causal representation learning highlighted in recent literature [10,29,41]. Additionally,

considering the diversity between image data and text data, where information in image data is typically much more details while information in text data tends to be more logically structured nature, we posit distinct causal mechanisms for the generation processes. Our causal generative model is formulated as following structural causal models [2]:

$$\mathbf{s} := \mathbf{g_s}(\mathbf{c}), \ \mathbf{x} := \mathbf{g_x}(\mathbf{c}, \mathbf{s}), \ \mathbf{t} := \mathbf{g_t}(\mathbf{c}, \mathbf{s}), \ \mathbf{y} := \mathbf{g_y}(\mathbf{c}). \tag{1}$$

In Eq. (1), the style variable $\mathbf{s}$ is causally influenced by the content via $\mathbf{g_s}$; $\mathbf{x}$ and $\mathbf{t}$ denote visual and textual data, respectively. Both visual and textual data are causally produced by the shared latent variables $\mathbf{c}$ and $\mathbf{s}$ through distinct, reversible generative processes: $\mathbf{g_x}$ for images and $\mathbf{g_t}$ for text data, respectively. The label $\mathbf{y}$ of a sample is exclusively determined by the content variable $\mathbf{c}$ via $\mathbf{g_y}$. For simplicity, exogenous noises are implicitly assumed but not explicitly represented in the causal generative model's formulation, aligning with the common understanding that each latent variable is influenced by exogenous noise.

Recent seminal work in [41] has demonstrated that the latent content variable $\mathbf{c}$ can be identified up to block identifiability (i.e., $\mathbf{c}$ can be isolated from style variable $\mathbf{s}$), by requiring all latent style variables to change (e.g., soft interventions on all latent style variables). This change can be achieved through image augmentation, i.e., the augmented image $\tilde{\mathbf{x}}$ can be interpreted as a generative result of $\tilde{\mathbf{s}}$, which is produced through soft interventions on original latent style variables $\mathbf{s}$. Despite such theoretical advancement, the practical implementation of this theoretical result within CLIP-like models remains unclear. In this study, we propose a practical method to disentangle content and style information within CLIP-like models by employing image augmentation, as detailed in Sec. 4.1. Moreover, we recognize that implementing sufficient changes on all latent style variables $\mathbf{s}$ through text augmentation is more feasible than image augmentation, due to high semanticity and logical structure in text data, we delve into the use of text augmentation to separate content information from style information, as discussed in Sec. 4.2.

4   Isolating Content from Style with Data Augmentation

In this section, we propose the employment of data augmentation to extract content information from the learned features in pre-trained CLIP-like models. Essentially, data augmentation facilitates the alteration of style factors while preserving content factors. Consequently, leveraging contrastive learning enables the segregation of content information from style information. We delve into two distinct forms of data augmentation, namely image augmentation (Sec. 4.1) and text augmentation (Sec. 4.2).

4.1   Isolating Content from Style with Augmented Images

While recent studies (von et al., 2021) have offered assurance regarding the disentanglement of content and style through contrastive learning with data augmentation, it remains unclear how these theoretical findings can be applied to
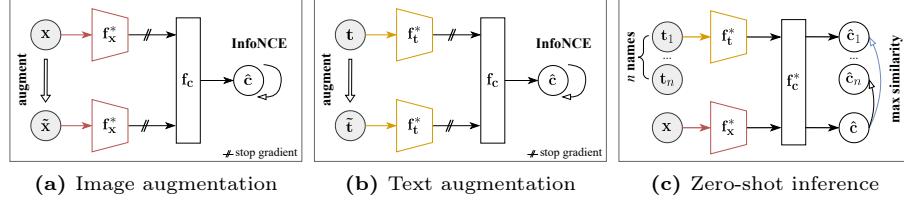
(a) Image augmentation   **(b)** Text augmentation   **(c)** Zero-shot inference

Fig. 2: Refining CLIP through data augmentation. (a) Training involves a disentangled network $\mathbf{f_c}$, utilizing contrastive loss on original and augmented image pairs $\mathbf{x}$ and $\tilde{\mathbf{x}}$, with CLIP's image encoder $\mathbf{f_x^*}$ holding frozen gradients. (b) More efficient content feature learning is achieved through contrastive learning with augmented text prompts $\mathbf{t}$ and $\tilde{\mathbf{t}}$, using the fixed text encoder $\mathbf{f_t^*}$ of CLIP. (c) Inference stage: The trained disentangled network $\mathbf{f_c^*}$ integrates with CLIP's text and image encoders, $\mathbf{f_t^*}$ and $\mathbf{f_x^*}$, to enable zero-shot inference for an input image $\mathbf{x}$ and class names $\mathbf{t}_1$ to $\mathbf{t}_n$.

the realm of vision-language models. We convert the theoretical findings into CLIP-like models in the following. The theoretical findings suggest using In-foNCE loss [34] to extract content information, as outlined below:

$$\mathcal{L}(\mathbf{f}; \{\mathbf{x}_i, \tilde{\mathbf{x}}_i\}_{i=1}^{b}, \tau) = -\frac{1}{b} \sum_{i=1}^{b} \log \frac{\exp\left[\langle \mathbf{f}(\mathbf{x}_i), \mathbf{f}(\tilde{\mathbf{x}}_i) \rangle / \tau\right]}{\sum_{j=1}^{b} \exp\left[\langle \mathbf{f}(\mathbf{x}_i), \mathbf{f}(\tilde{\mathbf{x}}_j) \rangle / \tau\right]}, \qquad (2)$$

where $\{\mathbf{x}_i\}_{i=1}^{b}$ represents a batch of b samples from the training dataset, $\mathbf{f}(\mathbf{x}_i)$ denotes sample $\mathbf{x}_i$'s features through model $\mathbf{f}$, $\tilde{\mathbf{x}}_i$ is the augmented counterpart of $\mathbf{x}_i$, and $\langle \mathbf{z}_1, \mathbf{z}_2 \rangle$ represents the cosine similarity between two feature vectors, $\mathbf{z}_1$ and $\mathbf{z}_2$, and τ represents the temperature parameter influencing the loss.

We extend it to refine pre-trained vision-language models, utilizing contrastive learning with augmented images (hereinafter referred to as "**Im.Aug**"). As illustrated in Fig. 2a, we train a disentangled network on top of CLIP's pre-trained image encoder. To enhance training efficiency and the usability of the proposed method, we freeze the pre-trained image encoder. Based on an InfoNCE loss, the learning objective of Im.Aug is formulated as follows:

$$\mathbf{f_c^*} = \operatorname*{argmin}_{\mathbf{f_c}} \mathbb{E}_{\{\mathbf{x}_i\}_{i=1}^{b} \in \mathcal{D}_{\mathbf{x}}} \mathcal{L}(\mathbf{f_c} \circ \mathbf{f_x^*}; \{\mathbf{x}_i, \tilde{\mathbf{x}}_i\}_{i=1}^{b}, \tau), \qquad (3)$$

where $\mathcal{D}_{\mathbf{x}}$ denotes the training image dataset and b represents the batch size, $\mathbf{f_c}$ is the disentangled network undergoing training. The pre-trained CLIP image encoder is represented by $\mathbf{f_x^*}$, with the asterisk "*" signifying that the model weights remain fixed. The variable $\mathbf{x}_i$ refers to an image sampled from $\mathcal{D}_{\mathbf{x}}$, and $\tilde{\mathbf{x}}_i$ is its augmented view.

The composition of the training dataset $\mathcal{D}_{\mathbf{x}}$, the image augmentation techniques used, the structure of the disentangled network $\mathbf{f_c}$, and the utilization of $\mathbf{f_c^*}$ post-training are detailed in the following subsections.

Data Synthesis and Image Augmentation To generate training image data, we combine class names with various image and object attributes to create text

Table 1: Template-based prompts. Attributes used to generate text prompts follow the structured format "a [art style] [image type] of a [object size] [object color] [class]", where "[class]" represents the class names.

Object Color	Object Size	Image Type	Art Style
yellow, green, black, blue, multicolored, orange, red, white, brown, purple	large, small, normal sized	painting, cartoon, infograph, sketch, photograph, clipart, mosaic art, sculpture	realistic, impressionistic

prompts for each class. Using a stable diffusion model [39], we produce synthetic images that comprise our training dataset $\mathcal{D}\mathbf{x}$. The creation of template prompts for stable diffusion is based on attributes such as object size, color, image type, and art style. As detailed in Tab. 1, the attributes include 10 colors and 3 sizes for objects, and 8 types and 2 art styles for images. By assembling these attributes into prompts like "a [art style] [image type] of a [object size] [object color] [class]", we generate 480 unique texts for each class, from which one image per prompt is synthesized. Further details on image synthesis and examples are available in Appendix B.1. For the image augmentation procedures, we adopt techniques commonly used in contrastive learning practice [7, 8, 41], specifically random cropping and color distortion.

Disentangled Network Structure Since the training process is based on CLIP's pre-trained lower-dimensional features, our disentangled network adopts a multi-layer perceptron (MLP) architecture. To fully benefit from the pre-trained CLIP text encoder, we construct a residual MLP featuring a zero-initialized projection, acting as the disentangled network, as depicted in Fig. 3. This design enables learning directly from the pre-trained representation space, avoiding a random starting point, inspired by ControlNet's zero-conv operation [46], which we adapt to a zero-linear operation within our residual MLP.

Within this architecture, the main branch includes a zero-initialized, bias-free linear layer positioned subsequent to the combination of a SiLU activation and a normally initialized linear layer. Conventionally, the dimensions of features before the initial linear layer, situated between the first and second linear layers, and following the second linear layer, are named as the input d_{in}, latent d_{mid}, and output d_{out} dimensions, respectively. To rectify any mismatches between the input and output dimensions, the network employs nearest-neighbor downsampling within the shortcut path, thereby ensuring both alignment and the preservation of sharpness for the input features. During the inference stage, a weighting parameter $\alpha > 0$ is introduced to modulate the portion of features emanating from the main branch before their integration with the input features, whereas this parameter remains constant at 1 throughout the training phase.

Inference After training, the disentangled network $\mathbf{f_c^*}$ is utilized following CLIP's image encoder to extract visual content features. Moreover, given that
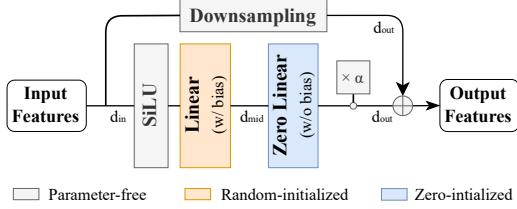
Fig. 3: Structure of the disentangled network. The architecture encompass a residual block featuring a zero-initialized, bias-free linear layer to commence optimization from the input feature space. When the input and output dimension differ, a downsampling operation is utilized to achieve alignment. During inference, a scalar parameter α balance the main branch and input features before combination.

vision-language data generation is rooted in a unified latent space, as depicted in Sec. 3, $\mathbf{f_c^*}$ can be seamlessly integrated with CLIP's image and text encoders to enhance zero-shot capabilities. As shown in Fig. 2c, for an image $\mathbf{x}$, the operation is formulated as the composition function $\mathbf{f_c^*} \circ \mathbf{f_x^*}(\mathbf{x})$, and similarly, for a text $\mathbf{t}$, as $\mathbf{f_c^*} \circ \mathbf{f_t^*}(\mathbf{t})$. This integration preserves CLIP's zero-shot functionality while achieving refined features through the improved disentanglement of content.

4.2 Isolating Content from Style with Augmented Prompts

Despite progress in disentangling content and style via image augmentation, adequately altering all style factors in an image remains challenging due to the high dimensionality and complexity of style information in images. Achieving substantial style changes through augmentation, essential for complete disentanglement [41], is difficult with existing image augmentation techniques. On the contrary, text data inherently possesses high semanticity and logical structure, making it more amenable to property-wise manipulation compared to image data. To further exploring the disentanglement of content, we propose Contrastive Learning with Augmented Prompts (**CLAP**).

   As depicted in Fig. 2b, CLAP employs an InfoNCE loss to train a disentangled network atop CLIP's pre-trained text encoder, keeping the encoder's gradients fixed, similar to Im.Aug. Leveraging the simpler structure of text, the template-based prompts previously utilized for synthesizing images now serve as the training text dataset, denoted by $\mathcal{D}_\mathbf{t}$. Utilizing the same disentangled network as in Im.Aug, the learning objective of CLAP is outlined as follows:

$$\mathbf{f_c^*} = \underset{\mathbf{f_c}}{\operatorname{argmin}} \; \underset{\{\mathbf{t}_i\}_{i=1}^b \in \mathcal{D}_\mathbf{t}}{\mathbb{E}} \; \mathcal{L}(\mathbf{f_c} \circ \mathbf{f_t^*}; \{\mathbf{t}_i, \tilde{\mathbf{t}}_i\}_{i=1}^b, \tau) + \lambda \mathcal{L}(\mathbf{f_c} \circ \mathbf{f_t^*}; \{\mathbf{t}_i^c, \tilde{\mathbf{t}}_i\}_{i=1}^b, 1), \quad (4)$$

where $\mathbf{f_t^*}$ denotes the pre-trained CLIP text encoder. The term $\mathbf{t}_i$ references a text prompt from $\mathcal{D}\mathbf{t}$, and $\tilde{\mathbf{t}}_i$ represents its augmented view, produced via prompt augmentation techniques. On the equation's right side, $\mathbf{t}_i^c$ specifies the class name associated with the text prompt $\mathbf{t}_i$. This strategy aims to enhance variations between prompt pairs, especially in cases where the text dataset $\mathcal{D}_\mathbf{t}$

Table 2: Prompt augmentation techniques. Various augmented views are generated from an original text prompt using specific augmentation techniques: OSD (Object Size Deletion), OCD (Object Color Deletion), ITD (Image Type Deletion), ASD (Art Style Deletion), and SPO (Swapping Prompt Order).

Original	OSD	OCD	ITD	ASD	SPO
a realistic painting of a large red car	a realistic painting of a red car	a realistic painting of a large car	a realistic of a large red car	a painting of a large red car	a large red car in a realistic painting

has a very limited number of samples. Here, λ serves for adjusting the second term's importance in the total loss function. All other symbols in Eq. (4) maintain their definitions as described earlier.

After training, the learned disentangled network is seamlessly integrated with both of CLIP's encoders to extract content representations, as depicted in Fig. 2c.

Prompt Augmentation To ensure text prompts undergo stylistic changes without compromising their content, we have developed specific augmentation techniques for synthetic text prompts. Drawing inspiration from Easy Data Augmentation (EDA) techniques [42], we adapted the Random Deletion (RD) and Random Swap (RS) techniques from EDA, customizing them to suit our prompt structure. To avoid inadvertently altering the content by introducing new object names or changing the core idea of a text prompt, our augmentation methods do not include random word insertions or replacements. Our primary augmentation techniques are Object Size Deletion (OSD), Object Color Deletion (OCD), Image Type Deletion (ITD), Art Style Deletion (ASD), and Swapping Prompt Order (SPO), each applied with a certain probability, as detailed in Tab. 2. Additionally, for down-stream datasets with few categories, to rich the population of training samples, we use an additional augmentation, named IGN (Inserting Gaussian Noise). Following the initializing protocol of prompt learning methods [47, 48], we insert a zero-mean Gaussian noise with 0.02 standard deviation with a noise length equals to 4, to the tokenized prompts.

Intuitively, these prompt augmentation methods parallel random masking techniques used in image augmentation [6, 17]. However, prompt augmentations are more effective and precise than their image counterparts. This effectiveness arises because prompt augmentations can specifically target and eliminate a particular style element without impacting the content, whereas image masking, operating at the pixel or patch level, might inadvertently damage content information or lead to insufficient style changes.

5   Experiments

We conduct three primary experiments to assess our method: (1) zero-shot evaluation with diverse prompts to gauge zero-shot performance and its robustness

to prompt perturbations; (2) linear probe tests on few-shot samples to evaluate the efficacy of the learned representations in few-shot settings; and (3) adversarial attack assessments on zero-shot and one-shot classifiers to determine their resistance to adversarial threats. We further conduct an ablative study on hyperparameters, explore the impact of different prompt augmentation combinations and various sources of training prompts on CLAP's performance, and replicate experiments across different CLIP model sizes.

5.1   Experimental Setup

Implementation. Im.Aug and CLAP are implemented using the ViT-B/16 CLIP model and executed on an NVIDIA RTX 3090 GPU. To ensure reproducibility, the random seed for all stochastic processes is fixed at 2024. More information on implementation details is provided in Appendix A.1.

Datasets. CLAP is assessed across four multi-domain datasets to examine its performance in varied environments: PACS [23] (4 domains, 7 categories), VLCS [1] (4 domains, 5 categories), OfficeHome [37] (4 domains, 65 categories), and DomainNet [35] (6 domains, 345 categories). For conciseness, we present average results across the domains for each dataset. Detailed experimental outcomes for each domain within these datasets are provided in Appendix A.4.

Compute efficiency. CLAP demonstrates faster convergence and shorter training times compared to Im.Aug. For CLAP, training on the PACS and VLCS datasets is completed in roughly 11 minutes, OfficeHome in approximately 14 minutes, and DomainNet in about 47 minutes. In contrast, Im.Aug requires around 16 minutes for PACS and VLCS, 50 minutes for OfficeHome, and 3.3 hours for DomainNet. Both Im.Aug and CLAP maintain CLIP's inference efficiency due to the disentangled network's efficient two-layer MLP structure.

5.2   Main Results

Zero-Shot Performance To assess zero-shot capabilities, CLAP undergoes evaluation using three specific fixed prompts: ZS(C), utilizing only the class name within "[class]"; ZS(PC), with the format "a photo of a [class]"; and ZS(CP), structured as "a [class] in a photo". To thoroughly examine zero-shot proficiency, a dynamic prompt, ZS(NC), formatted as "[noise][class]", is also used, where "[noise]" signifies the introduction of Gaussian noise characterized by a mean of 0 and a standard deviation of 0.02.

As presented in Tab. 3, CLAP surpasses both CLIP and Im.Aug across all evaluated prompts for every dataset. Unlike the uniform enhancement in zero-shot performance CLAP achieves over CLIP, Im.Aug displays inconsistent results. A closer examination reveals CLAP's superiority over CLIP is especially significant for the dynamic ZS(NC) prompt. This demonstrates CLAP's effectiveness in significantly improving zero-shot performance compared to the original CLIP representations.

In assessing the model's robustness to prompt perturbations, we examine the variances in zero-shot performance across different prompts by analyzing the

Table 3: Zero-shot results across three distinct prompts: "C" for "[class]", "CP" for "a [class] in a photo", "PC" for "a photo of a [class]", and a dynamic prompt "NC" for "[noise][class]" showcase that CLAP consistently outperforms CLIP's zero-shot performance across all datasets, whereas image augmentation exhibits mixed outcomes.

Prompt	Method	Zero-shot performance, avg. top-1 acc. (%) ($\uparrow$)				
		PACS	VLCS	Off.Home	Dom.Net	Overall
ZS(C)	CLIP	95.7	76.4	79.8	57.8	77.4
	Im.Aug	96.5	79.5	77.0	51.5	76.1
	CLAP	**97.2**	**82.6**	**81.0**	**58.7**	**79.9**
ZS(CP)	CLIP	95.2	82.0	79.5	57.0	78.4
	Im.Aug	96.3	82.9	75.8	50.7	76.4
	CLAP	**97.3**	**83.4**	**80.5**	**58.0**	**79.8**
ZS(PC)	CLIP	96.1	82.4	82.5	57.7	79.7
	Im.Aug	96.5	83.0	78.6	51.6	77.4
	CLAP	**97.2**	**83.4**	**83.0**	**59.0**	**80.6**
ZS(NC)	CLIP	90.8	68.3	71.5	51.0	70.4
	Im.Aug	94.8	73.1	67.5	44.0	69.9
	CLAP	**97.2**	**81.0**	**73.5**	**52.6**	**76.1**

range (R) and standard deviation (δ) of results derived from ZS(C), ZS(CP), and ZS(PC). Additionally, we investigate the decline ($\Delta_{(NC)}$) in performance from ZS(C) to ZS(NC) as a broad indicator of resilience to noised prompts.

As presented in Tab. 4, CLAP significantly reduces the variance in zero-shot performance across various testing prompts, evidenced by markedly lower values of δ and R, and a less pronounced decrease in performance with a noised prompt, in contrast to Im.Aug and the baseline representations of CLIP. Although Im.Aug aids in reducing performance variance to some extent, its efficacy is notably inferior to that of CLAP. These findings highlight CLAP's enhanced robustness in maintaining consistent zero-shot performance across a diverse array of prompts.

Few-Shot Performance We conduct evaluations of 1-shot, 4-shot, 8-shot, and 16-shot linear probes across each domain within the four datasets. As illustrated in Fig. 4, CLAP significantly outperforms both CLIP and Im.Aug in few-shot learning scenarios. Notably, in the 1-shot setting CLAP achieves performance improvements over the linear-probe CLIP model by margins of +10%, +3.5%, +2.5%, and +1.5% on the PACS, VLCS, OfficeHome, and DomainNet datasets, respectively. These improvements are especially significant in comparison to the gains observed with Im.Aug counterparts, underpinning CLAP's efficacy in few-shot scenarios. For detailed quantitative results, please refer to Appendix A.4.

Adversarial Performance To assess adversarial robustness, zero-shot (ZS(C)) and one-shot classifiers are evaluated against prominent adversarial attack methods, such as FGSM [15], PGD [30], and CW [5], by generating adversarial sam-

Table 4: CLAP more effectively reduces zero-shot performance variance across prompts than image augmentation, with R and δ indicating the range and standard deviation for ZS(C), ZS(CP), and ZS(PC). The decrease $\Delta_{(NC)}$ from ZS(C) to ZS(NC) highlights CLAP's enhanced robustness against prompt perturbations.

Metric	Method	Performance variance, avg. top-1 acc. (%) ($\downarrow$)				
		PACS	VLCS	Off.Home	Dom.Net	Overall
	CLIP	0.9	6.1	3.1	**0.8**	2.7
R	Im.Aug	**0.1**	3.6	2.8	0.9	1.9
	CLAP	**0.1**	**0.8**	**2.5**	1.0	**1.1**
	CLIP	0.4	2.8	1.4	**0.4**	1.2
δ	Im.Aug	0.1	1.7	1.2	**0.4**	0.8
	CLAP	**0.0**	**0.4**	**1.1**	**0.4**	**0.5**
	CLIP	4.9	8.1	8.3	6.8	7.0
$\Delta_{(NC)}$	Im.Aug	1.6	6.4	9.5	7.5	6.3
	CLAP	**0.0**	**1.6**	**7.5**	**6.1**	**3.8**

Fig. 4: Few-shot linear probe comparisons of image-encoder features show that CLAP enhances CLIP's few-shot performance more effectively than Im.Aug. In the accompanying figure, "ZS" indicates the zero-shot performance using a "[class]" prompt.

ples for testing. For FGSM, 1 adversarial iteration is employed, whereas for PGD and CW, 20 iterations are used, all with an epsilon of 0.031. As indicated in Tab. 5, classifiers utilizing CLAP representations demonstrate superior resilience to these adversarial attacks compared to those based on CLIP representations. Across the four datasets, CLAP's zero-shot and 1-shot classifiers surpass CLIP by margins of $+7.6\%$ and $+8.5\%$ against FGSM, $+1.0\%$ and $+11.7\%$ against PGD-20, and $+1.1\%$ and $+2.3\%$ against CW-20, respectively. These figures notably exceed the performance improvements of $+4.4\%$ and $+4.6\%$ against FGSM, $+0.3\%$ and $+6.2\%$ against PGD-20, and 0% and $+1.3\%$ against CW-20 achieved by Im.Aug. The result suggests that CLAP efficiently enhances robustness against adversarial attacks in both zero-shot and one-shot scenarios.

5.3   More Analysis

t-SNE Visualization In our t-SNE visualizations, we examine the representations of CLIP, Im.Aug, and CLAP for all images within the Art Painting domain of the PACS dataset. Fig. 5 shows that CLAP's image representations display a marked inter-class separation and tighter intra-class clustering than those of

Table 5: Image augmentation and CLAP both enhance CLIP's zero-shot with the "[class]" prompt and 1-shot robustness against adversarial attacks, with CLAP showing greater improvements.

Setting	Method	Avg. top-1 acc. (%) under adversarial attacks(↑)												
		FGSM				PGD-20				CW-20				Avg.
		PACS	VLCS	O.H.	D.N.	PACS	VLCS	O.H.	D.N.	PACS	VLCS	O.H.	D.N.	
	CLIP	86.8	65.6	57.9	22.5	29.1	2.0	10.1	10.7	27.4	1.5	7.4	7.6	29.2
ZS(C)	Im.Aug	88.0	69.6	55.1	37.9	**31.3**	2.1	10.4	9.0	29.4	1.7	7.0	5.8	31.1
	CLAP	**88.7**	**71.9**	**58.5**	**44.2**	30.8	**3.2**	**10.6**	**11.2**	**29.8**	**2.3**	**8.1**	**8.0**	**32.7**
	CLIP	66.7	45.2	34.3	22.5	34.8	16.0	5.6	11.3	18.9	0.7	4.5	3.2	23.7
1-shot	Im.Aug	79.4	47.1	**37.1**	23.5	55.2	16.1	**8.5**	**12.5**	23.2	0.9	**5.1**	3.4	28.0
	CLAP	**89.6**	**52.2**	**37.1**	**23.9**	**73.4**	**21.2**	7.4	**12.5**	**27.0**	1.1	5.0	**3.5**	**31.9**

(a) CLIP   (b) Im.Aug   (c) CLAP

Fig. 5: t-SNE visualizations of all images in the Art Painting of PACS dataset show CLAP outperforms the original CLIP and Im.Aug, with clearer inter-class distinctions and tighter intra-class clusters.

CLIP and Im.Aug. This observation suggests that CLAP's representations are more closely tied to content information and less influenced by style information, in contrast to the other two.

Ablations In Fig. 6, we assess the zero-shot capabilities of our model using two distinct prompts, ZS(C) and ZS(PC), on the VLCS dataset. This analysis forms part of an ablative study aimed at understanding the influence of various hyper-parameters on model performance. Specifically, we examine: the dimensions of the latent layer within the MLP of the disentangled network, as illustrated in Fig. 6a; the temperature parameter (τ) in the loss function, as depicted in Fig. 6b; and the weight coefficient (α) during the inference stage, as shown in Fig. 6c. Our findings indicate that CLAP consistently enhances zero-shot performance across all tested configurations for both prompts, while also significantly reducing the gap between the performances elicited by each prompt. These results underscore the efficacy of CLAP in accommodating a wide range of hyper-parameters.

Prompt Augmentation Combinations We explore diverse combinations of our tailored prompt augmentation methods and examine Easy Data Augmentation (EDA) techniques [42] on the VLCS dataset. Each tested technique showcases CLAP's enhancements over CLIP, with details available in Appendix A.2.

(a) Latent dimensions   **(b)** Temperature τ in loss   **(c)** Inference weight $\log_{10}(\alpha)$

Fig. 6: We conduct ablative study on hyper-parameter choices on the VLCS dataset, including latent dimensions, τ values, and α values during the inference stage. CLAP continuously enhance CLIP's performance throughout the tested values.
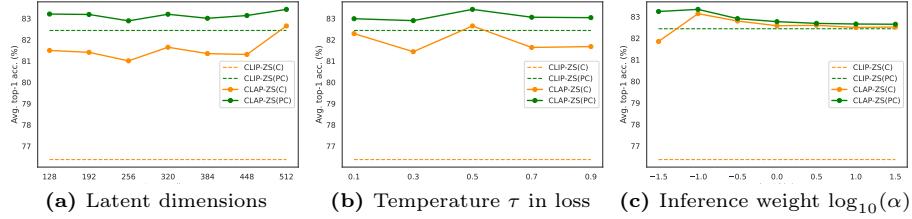
Prompt Sources We assess the impact of different training prompt formats, originating from various synthetic sources, on the performance of the VLCS dataset, incorporating EDA techniques. Our evaluation includes our template-based prompts, LLM-generated prompts by ChatGPT-3.5 [3] (with the generation process detailed in Appendix B.2), prompts structured as "a [random] style of [class]," where "[random]" is filled with terms from a random word generator[2], and prompts produced using the PromptStyler method [9]. The findings indicate that the training prompts with simpler forms tend to yield better performance, with detailed quantitative results presented in Appendix A.3.

Experiments on Different Model Scales In our repeated experiments assessing zero-shot performance on the ViT-L/14 and ResNet50x16 pre-trained with CLIP, we consistently find that CLAP improves zero-shot performance while also reducing performance variances. This consistent observation underscores CLAP's effectiveness in enhancing the quality of CLIP representations. For quantitative details supporting these findings, please see the Appendix C.

6   Conclusion

To enhance pre-trained CLIP-like models, this study delves into disentangling latent content variables. Through a causal analysis of the underlying generative processes of vision-language data, we discover that training a disentangled network in one modality can effectively disentangle content across both modalities. Given the high semantic nature of text data, we identify that disentanglement is more achievable within the language modality through text augmentation interventions. Building on these insights, we introduce CLAP (Contrastive Learning with Augmented Prompts) to acquire disentangled vision-language content features. Comprehensive experiments validate CLAP's effectiveness, demonstrating significant improvements in zero-shot and few-shot performance, and enhancing robustness against perturbations. We anticipate that our work will inspire further exploration into disentangling latent variables within vision-language models.

---

[2] https://github.com/vaibhavsingh97/random-word

References

1. Albuquerque, I., Monteiro, J., Darvishi, M., Falk, T.H., Mitliagkas, I.: Generalizing to unseen domains via distribution matching. arXiv preprint arXiv:1911.00804 (2019). `https://doi.org/10.48550/arXiv.1911.00804`
2. Bollen, K.A.: Structural equations with latent variables, vol. 210. John Wiley & Sons (1989). `https://doi.org/10.1002/9781118619179`
3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020), `https://dl.acm.org/doi/abs/10.5555/3495724.3495883`
4. Carlini, N., Terzis, A.: Poisoning and backdooring contrastive learning. In: International Conference on Learning Representations (2021), `https://openreview.net/forum?id=iC4UHbQ01Mp`
5. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP). pp. 39–57. IEEE Computer Society (2017). `https://doi.org/10.1109/SP.2017.49`
6. Chen, P., Liu, S., Zhao, H., Jia, J.: Gridmask data augmentation. arXiv preprint arXiv:2001.04086 (2020). `https://doi.org/10.48550/arXiv.2001.04086`
7. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020), `https://dl.acm.org/doi/abs/10.5555/3524938.3525087`
8. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15750–15758 (2021). `https://doi.org/10.1109/CVPR46437.2021.01549`
9. Cho, J., Nam, G., Kim, S., Yang, H., Kwak, S.: Promptstyler: Prompt-driven style generation for source-free domain generalization. In: Proceedings of the IEEE international conference on computer vision. pp. 15702–15712 (2023). `https://doi.org/10.1109/ICCV51070.2023.01439`
10. Daunhawer, I., Bizeul, A., Palumbo, E., Marx, A., Vogt, J.E.: Identifiability results for multimodal contrastive learning. ICLR (2023), `https://openreview.net/forum?id=U_2kuqoTcB`
11. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2020), `https://openreview.net/forum?id=YicbFdNTTy`
12. Fort, S.: Adversarial vulnerability of powerful near out-of-distribution detection. arXiv preprint arXiv:2201.07012 (2022). `https://doi.org/10.48550/arXiv.2201.07012`
13. Gao, P., Geng, S., Zhang, R., Ma, T., Fang, R., Zhang, Y., Li, H., Qiao, Y.: Clip-adapter: Better vision-language models with feature adapters. International Journal of Computer Vision **132**(2), 581–595 (2024). `https://doi.org/10.1007/s11263-023-01891-x`
14. Ge, C., Huang, R., Xie, M., Lai, Z., Song, S., Li, S., Huang, G.: Domain adaptation via prompt learning. arXiv preprint arXiv:2202.06687 (2022). `https://doi.org/10.48550/arXiv.2202.06687`
15. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: International Conference on Learning Representations (2015). `https://doi.org/10.48550/arXiv.1412.6572`

16. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Advances in neural information processing systems **33**, 21271–21284 (2020), `https://dl.acm.org/doi/abs/10.5555/3495724.3497510`

17. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16000–16009 (2022). `https://doi.org/10.1109/CVPR52688.2022.01553`

18. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020). `https://doi.org/10.1109/CVPR42600.2020.00975`

19. Hong, T., Guo, X., Ma, J.: Itmix: Image-text mix augmentation for transferring clip to image classification. In: 2022 16th IEEE International Conference on Signal Processing (ICSP). vol. 1, pp. 129–133. IEEE (2022). `https://doi.org/10.1109/ICSP56322.2022.9965292`

20. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: International conference on machine learning. pp. 4904–4916. PMLR (2021), `https://proceedings.mlr.press/v139/jia21b.html`

21. Khattak, M.U., Rasheed, H., Maaz, M., Khan, S., Khan, F.S.: Maple: Multi-modal prompt learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19113–19122 (2023). `https://doi.org/10.1109/CVPR52729.2023.01832`

22. Kong, L., Xie, S., Yao, W., Zheng, Y., Chen, G., Stojanov, P., Akinwande, V., Zhang, K.: Partial disentanglement for domain adaptation. In: International Conference on Machine Learning. pp. 11455–11472. PMLR (2022), `https://proceedings.mlr.press/v162/kong22a.html`

23. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.M.: Deeper, broader and artier domain generalization. In: Proceedings of the IEEE international conference on computer vision. pp. 5542–5550 (2017). `https://doi.org/10.1109/ICCV.2017.591`

24. Li, H., Wang, X., Zhang, Z., Yuan, Z., Li, H., Zhu, W.: Disentangled contrastive learning on graphs. Advances in Neural Information Processing Systems **34**, 21872–21884 (2021), `https://dl.acm.org/doi/10.5555/3540261.3541935`

25. Li, Y., Liang, F., Zhao, L., Cui, Y., Ouyang, W., Shao, J., Yu, F., Yan, J.: Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. In: International Conference on Learning Representations (2021)

26. Liu, Y., Zhang, Z., Gong, D., Gong, M., Huang, B., van den Hengel, A., Zhang, K., Shi, J.Q.: Identifiable latent polynomial causal models through the lens of change. In: The Twelfth International Conference on Learning Representations (2024), `https://openreview.net/forum?id=ia9fKO1Vjq`

27. Liu, Y., Zhang, Z., Gong, D., Gong, M., Huang, B., Hengel, A.v.d., Zhang, K., Shi, J.Q.: Identifying weight-variant latent causal models. arXiv preprint arXiv:2208.14153 (2022). `https://doi.org/10.48550/arXiv.2208.14153`

28. Liu, Y., Zhang, Z., Gong, D., Gong, M., Huang, B., Hengel, A.v.d., Zhang, K., Shi, J.Q.: Identifiable latent neural causal models. arXiv preprint arXiv:2403.15711 (2024). `https://doi.org/10.48550/arXiv.2403.15711`

29. Liu, Y., Zhang, Z., Gong, D., Gong, M., Huang, B., Zhang, K., Shi, J.Q.: Identifying latent causal content for multi-source domain adaptation. arXiv preprint arXiv:2208.14161 (2022). `https://doi.org/10.48550/arXiv.2208.14161`

30. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (2018), `https://openreview.net/forum?id=rJzIBfZAb`

31. Mahajan, D., Tople, S., Sharma, A.: Domain generalization using causal matching. In: International Conference on Machine Learning. pp. 7313–7324. PMLR (2021), `https://proceedings.mlr.press/v139/mahajan21b.html`

32. Mamooler, S.: Clip explainability. `https://github.com/sMamooler/CLIP_Explainability`, accessed: 2024-03-06

33. Mao, C., Geng, S., Yang, J., Wang, X., Vondrick, C.: Understanding zero-shot adversarial robustness for large-scale models. In: The Eleventh International Conference on Learning Representations (2022), `https://openreview.net/forum?id=P4bXCawRi5J`

34. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018). `https://doi.org/10.48550/arXiv.1807.03748`

35. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: Proceedings of the IEEE international conference on computer vision. pp. 1406–1415 (2019). `https://doi.org/10.1109/ICCV.2019.00149`

36. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021), `https://proceedings.mlr.press/v139/radford21a.html`

37. Rahman, M.M., Fookes, C., Baktashmotlagh, M., Sridharan, S.: Multi-component image translation for deep domain generalization. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 579–588. IEEE (2019). `https://doi.org/10.1109/WACV.2019.00067`

38. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: International Conference on Machine Learning. pp. 8821–8831. PMLR (2021), `https://proceedings.mlr.press/v139/ramesh21a.html`

39. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022). `https://doi.org/10.1109/CVPR52729.2023.01389`

40. Sanchez, E.H., Serrurier, M., Ortner, M.: Learning disentangled representations via mutual information estimation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16. pp. 205–221. Springer (2020). `https://doi.org/10.1007/978-3-030-58542-6_13`

41. Von Kügelgen, J., Sharma, Y., Gresele, L., Brendel, W., Schölkopf, B., Besserve, M., Locatello, F.: Self-supervised learning with data augmentations provably isolates content from style. Advances in neural information processing systems **34**, 16451–16467 (2021), `https://dl.acm.org/doi/10.5555/3540261.3541519`

42. Wei, J., Zou, K.: Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 6382–6388 (2019). `https://doi.org/10.18653/v1/D19-1670`

43. Wortsman, M., Ilharco, G., Kim, J.W., Li, M., Kornblith, S., Roelofs, R., Lopes, R.G., Hajishirzi, H., Farhadi, A., Namkoong, H., et al.: Robust fine-tuning of zero-shot models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7959–7971 (2022). `https://doi.org/10.1109/CVPR52688.2022.00780`

44. Yang, M., Liu, F., Chen, Z., Shen, X., Hao, J., Wang, J.: Causalvae: Disentangled representation learning via neural structural causal models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9593–9602 (2021). `https://doi.org/10.1109/CVPR46437.2021.00947`

45. Yang, W., Mirzasoleiman, B.: Robust contrastive language-image pretraining against adversarial attacks. arXiv preprint arXiv:2303.06854 (2023). `https://doi.org/10.48550/arXiv.2303.06854`

46. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: IEEE International Conference on Computer Vision (ICCV) (2023). `https://doi.org/10.1109/ICCV51070.2023.00355`

47. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16816–16825 (2022). `https://doi.org/10.1109/CVPR52688.2022.01631`

48. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. International Journal of Computer Vision **130**(9), 2337–2348 (2022). `https://doi.org/10.1007/s11263-022-01653-1`

49. Zimmermann, R.S., Sharma, Y., Schneider, S., Bethge, M., Brendel, W.: Contrastive learning inverts the data generating process. In: International Conference on Machine Learning. pp. 12979–12990. PMLR (2021), `https://proceedings.mlr.press/v139/zimmermann21a.html`

CLAP: Isolating Content from Style through Contrastive Learning with Augmented Prompts
APPENDIX

Yichao Cai ⓘ, Yuhang Liu ⓘ, Zhen Zhang ⓘ, and Javen Qinfeng Shi ⓘ

Australian Institute for Machine Learning, University of Adelaide, SA 5000, Australia
{yichao.cai,yuhang.liu01,zhen.zhang02,javen.shi}@adelaide.edu.au

Overview of the Appendix:

- More details on experiments using the CLIP pre-trained ViT-B/16 model are provided in Appendix A, including implementation details in Appendix A.1, investigations into prompt augmentation combinations in Appendix A.2, analysis of different training prompt sources in Appendix A.3, and detailed experiment results for each dataset in Appendix A.4.
- The processes of data synthesis with large models used in our approach are outlined in Appendix B: The image synthesis procedure for Im.Aug is detailed in Appendix B.1, and the approach for generating "LLM" prompts, used in analyzing prompt sources, is described in Appendix B.2.
- In Appendix C, we detail our repeated zero-shot experiments conducted with the CLIP pre-trained ViT-L/14 (Appendix C.1) and ResNet50x16 (Appendix C.2) models.
- In section Appendix D, we present discussions covering the underlying rationale for basing CLAP on the CLIP pre-trained models in Appendix D.1, and the impact of image augmentation and text augmentation in Appendix D.2.

A More on Experiments with ViT-B/16

A.1 Implementation Details

In this section, we detail the implementation of our experiments utilizing the CLIP pre-trained ViT-B/16 model:

Network. The network's output dimension is aligned with the 512-dimensional CLIP features, thereby obviating the need for input feature downsampling. The latent dimensions are tailored to each dataset: 256 for PACS, 448 for OfficeHome, and 512 for VLCS and DomainNet, to accommodate the variety of categories and complexity of datasets. The weight parameter α is adjusted to 0.208 for PACS, 0.056 for VLCS, 0.14 for OfficeHome, and 0.2 for DomainNet, while it is consistently maintained at 1 throughout the training phase.

Training CLAP. Training parameters are consistent across datasets, employing the Adam optimizer with a learning rate of 0.0001, limiting training to 8,000 steps with checking the average loss every 480 steps, and instituting early stopping after five checkpoints without a loss decrease of at least 0.01. Batch sizes are adjusted to 8 for PACS and VLCS, 96 for OfficeHome, and 384 for Domain-Net, with the temperature parameter τ set at 0.5 for PACS and VLCS, and 0.3 for OfficeHome and DomainNet. The loss coefficient λ is set to 1 for PACS and VLCS, and 0.0001 for OfficeHome and DomainNet, due to the first two datasets have less classes. Prompt augmentations, OSD+OCD+SPO, are applied across datasets all with a 0.5 probability. For the PACS and VLCS datasets, Gaussian noise with a zero mean and a standard deviation of 0.02 is randomly inserted at the beginning, middle, or end of the augmented-view prompts to enrich the training samples. In the linear probe evaluations for few-shot analysis, L2 normalization and cross-entropy loss are utilized for training over 1,000 epochs with a batch size of 32, incorporating early stopping with a patience threshold of 10 epochs and a loss decrease criterion of 0.001.

Training Im.Aug. We train a disentangled network using image augmentation, applying the InfoNCE loss with a temperature parameter τ set to 0.5. This include image augmentation techniques, image cropping ($scale \in [0.64, 1.0]$) and color distortion ($brightness = 0.5, hue = 0.3$), each with a probability of 0.5. Other training and inference configurations for Im.Aug are consistent with those used for CLAP across all datasets.

A.2   Prompt Augmentation Combinations

In Tab. 1, we explore different combinations of our tailored prompt augmentation techniques and EDA (Easy Data Augmentation) [42] techniques on the VLCS dataset. Each combination demonstrates CLAP's effectiveness in enhancing CLIP's performance and reducing performance disparities. The combination of OSD+OCS+SPO+IGN achieves the highest average accuracy and the least variance, outperforming the EDA techniques. Notably, even without incorporating random noise in the augmentations, CLAP significantly surpasses CLIP in handling perturbations on prompts, as evidenced by the largely reduced $\Delta_{(NC)}$.

A.3   Prompt Sources

In Tab. 2, we examine the effects of various training prompt formats, sourced from different synthetic origins, on the VLCS dataset performance, utilizing
    EDA techniques. The prompt formats are defined as follows: "Template" refers to the template-based prompts fundamental to our primary approach; "LLM" designates prompts created by ChatGPT-3.5 [3], with the generation process elaborated in Appendix B.2; "Random" describes prompts formatted as "a [random] style of [class]," with "[random]" being replaced by terms from a random word generator; and "Prm.Stl." indicates vectorized prompts generated through PromptStyler [9].

Table 1: We evaluate prompt augmentation combinations on the VLCS dataset: OSD (①), OCD (②), ITD (③), ASD (④), SPO (⑤), and IGN (⑥). ZS(Avg.) shows average zero-shot accuracy acoss four distinct inference prompts. CLAP boosts CLIP's accuracy and reduces variances, with ①②⑤⑥ as the optimal combination.

Metrics	CLIP (base)	Avg. top-1 acc. (%) of different augmentations						
		EDA	①②③④⑤⑥	①②③④⑤	①②③④⑥	①②③④	③④⑤⑥	①②⑤⑥
ZS(Avg.) (↑)	77.3	81.6	82.0	80.1	82.0	79.6	82.1	**82.6**
R (↓)	6.1	1.9	1.2	2.5	0.9	3.2	1.6	**0.8**
δ (↓)	2.8	0.9	0.6	1.2	**0.4**	1.5	0.7	**0.4**
$\Delta_{(NC)}$ (↓)	8.1	2.3	1.7	3.0	1.8	3.4	2.0	**1.6**

Table 2: We employ EDA augmentation to train CLAP with diverse prompt sources on the VLCS dataset. Each prompt source contributes to improvements in CLIP's zero-shot performance, with "Random" and "Template" prompts, in their simpler forms, yielding better outcomes.

Metrics	CLIP (base)	Avg. top-1 acc. (%) of different sources			
		LLM	Random	Prm.Stl.	Template
ZS(Avg.) (↑)	77.3	78.2	**81.6**	81.2	**81.6**
R (↓)	6.1	3.2	**0.7**	2.7	1.9
δ (↓)	2.8	1.5	**0.3**	1.2	0.9
$\Delta_{(NC)}$ (↓)	8.1	3.3	**2.3**	3.0	**2.3**

Our experimental results indicate that CLAP, when trained across these varied prompt formats, enhances the performance of CLIP. Notably, despite the complex generation mechanisms of "LLM" and "Prm.Stl." prompts, the simpler, random-styled and template-based prompts demonstrate superior efficacy. However, it is important to highlight that the improvements attributed to these diverse prompt formats, trained with EDA, do not surpass the best performance of the prompt augmentations tailored for template-based prompts.

A.4   Detailed Results on ViT-B/16

Details on Zero-Shot Evaluations We present the domain-level zero-shot performance with various prompts across each dataset in Tab. 3. CLAP consistently enhances CLIP's zero-shot performance across these different prompts. Given that CLAP exclusively utilizes text data for training, it does not compromise CLIP's inherent ability to generalize across domains, which is acquired from its extensive training dataset. Rather, by achieving a more effective disentanglement of content, it unequivocally enhances CLIP's zero-shot performance across all dataset domains.

Table 3: Domain-level zero-shot results of the ViT-B/16 model on the test datasets.

Dataset Domains		ZS(C)			ZS(CP)			ZS(PC)			ZS(NC)		
		CLIP	Im.Aug	CLAP	CLIP	Im.Aug	CLAP	CLIP	Im.Aug	CLAP	CLIP	Im.Aug	CLAP
PACS	A.	96.4	96.9	97.5	93.4	97.0	97.6	97.4	97.6	97.6	87.8	93.5	97.1
	C.	98.9	99.0	98.9	99.0	99.2	99.0	99.1	99.0	98.9	95.4	97.6	98.8
	P.	99.9	99.9	99.9	99.3	99.6	99.9	99.9	99.9	99.9	93.1	99.0	99.9
	S.	87.7	90.1	92.5	89.2	89.6	92.5	88.1	89.4	92.3	87.1	89.3	93.1
VLCS	C.	99.7	99.8	99.9	99.9	99.9	99.9	99.9	99.9	99.9	87.0	96.0	99.9
	L.	61.8	66.2	67.7	69.9	70.4	70.4	70.2	70.2	70.7	55.9	59.9	65.9
	S.	70.1	74.8	78.0	73.3	76.0	77.2	73.6	76.4	76.9	61.4	66.2	75.3
	V.	73.9	77.1	84.9	84.8	85.4	86.0	86.1	85.6	86.2	68.9	70.3	82.9
OfficeHome	A.	80.5	79.0	81.8	80.1	76.0	81.6	83.2	78.7	83.2	73.0	69.2	73.6
	C.	64.6	59.6	66.4	63.7	58.9	65.4	68.1	61.9	69.0	57.0	52.0	60.4
	P.	86.3	83.6	87.5	86.6	83.4	87.2	89.1	86.6	89.7	77.2	72.3	78.9
	R.	88.0	85.9	88.5	87.6	84.8	87.7	89.8	87.2	90.0	79.0	76.5	81.1
DomainNet	C.	71.0	64.3	71.9	70.5	62.1	72.0	71.3	63.4	72.8	63.2	53.9	64.6
	I.	48.6	40.5	50.6	47.7	40.7	49.5	47.8	40.0	50.5	42.9	35.0	45.1
	P.	66.6	59.1	67.7	66.0	59.0	67.3	66.5	59.8	68.4	57.2	50.4	59.4
	Q.	14.9	12.4	15.2	13.3	11.5	13.8	14.1	11.8	14.3	12.0	9.2	13.1
	R.	82.6	76.6	83.1	82.2	75.8	82.2	83.4	78.2	83.7	75.2	67.9	75.6
	S.	63.1	56.1	63.7	62.2	55.0	63.1	63.4	56.4	64.4	55.7	47.5	57.6

The header spanning row reads: Domain-level avg. top-1 acc. (%) of zero-shot performance usig ViT-B/16 (↑)

Details on Few-Shot Evaluations We display the quantitative results of few-shot performance in Tab. 4. CLAP consistently enhances the few-shot capabilities, showcasing improvements across test datasets at a closer domain level.

Details on Adversarial Evaluations In Tab. 5, we detail our adversarial performance evaluations for PACS, VLCS, OfficeHome, and DomainNet, respectively. CLAP enhances both zero-shot and one-shot performance across all domains of the tested datasets. While Im.Aug boosts one-shot robustness against adversarial tasks, its impact on zero-shot adversarial robustness is inconsistent.

Details on Ablative Analysis In Tab. 6, we provide detailed results from our analysis on zero-shot performance using various combinations of prompt augmentations. Additionally, in Tab. 7, we present the outcomes of our ablative studies focusing on the hyperparameters τ, latent dimension, and α, respectively, each evaluated domain-wise. The results indicate that CLAP is effective across a wide range of hyperparameters.
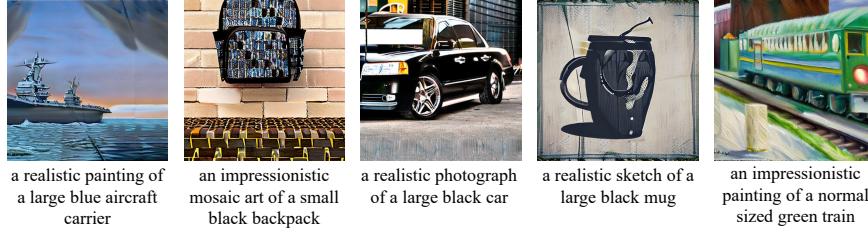
B   Data Synthesis

B.1   Synthetic Image Generation

We employ the stable diffusion [39] v2.1 model for generating synthetic images used in our comparing experiments, specifically utilizing the Stable Diffusion

Table 4: Domain-level few-shot results of the ViT-B/16 model using the test datasets.

Dataset	Domains	Domain-level avg. top-1 acc. (%) of few-shot performance of ViT-B/16 (↑)													
		1-shot			4-shot			8-shot			16-shot			32-shot		
		CLIP	Im.Aug	CLAP	CLIP	Im.Aug	CLAP	CLIP	Im.Aug	CLAP	CLIP	Im.Aug	CLAP	CLIP	Im.Aug	CLAP
PACS	A.	79.5	84.1	94.5	92.4	96.4	97.2	95.1	97.2	98.4	97.9	98.1	98.4	98.8	99.1	98.9
	C.	86.7	96.1	98.3	96.8	98.6	99.2	98.8	98.9	99.3	99.5	99.2	99.5	99.6	99.6	99.6
	P.	97.4	99.8	99.9	99.6	99.8	99.9	99.9	99.9	99.9	99.8	99.9	99.9	99.9	99.9	99.9
	S.	75.1	80.0	87.3	91.1	92.3	92.5	92.3	92.3	92.9	92.4	92.6	93.1	93.9	94.2	94.1
VLCS	C.	99.2	99.7	99.8	99.9	99.8	99.9	99.8	99.7	99.9	99.7	99.9	99.9	99.9	100.0	99.9
	L.	41.3	41.3	41.1	56.7	57.0	59.8	46.2	36.8	48.3	59.4	60.4	62.6	60.4	60.7	61.9
	S.	45.3	46.1	50.8	61.9	63.7	69.0	67.4	67.7	71.3	75.9	76.8	80.9	77.4	78.6	81.0
	V.	50.9	53.4	59.0	64.5	66.7	76.1	75.4	74.1	78.7	72.6	73.9	77.7	85.7	86.1	87.9
OfficeHome	A.	42.6	45.1	43.9	76.8	77.6	77.7	84.8	86.0	85.5	91.8	92.1	92.1	97.4	97.5	97.5
	C.	40.1	45.0	43.8	69.9	70.2	70.5	75.8	75.9	76.6	81.6	81.6	81.6	89.0	89.0	89.2
	P.	70.2	73.3	73.4	89.7	90.3	90.2	93.8	93.7	93.9	95.7	95.7	95.8	97.7	97.6	97.6
	R.	58.4	59.3	59.4	81.7	83.1	82.9	89.7	89.5	89.9	92.9	92.7	93.2	95.8	95.8	95.8
DomainNet	C.	42.1	43.6	43.8	66.8	67.5	67.8	74.2	74.3	74.6	78.5	78.6	78.8	82.8	82.8	82.7
	I.	19.5	20.8	21.0	38.5	39.3	39.7	46.7	47.0	47.3	53.2	53.2	53.6	60.0	59.9	60.1
	P.	32.1	33.5	34.2	60.5	60.9	61.5	68.0	68.0	68.7	72.5	72.6	73.0	76.7	76.6	76.8
	Q.	15.2	15.3	15.3	30.0	29.6	29.9	37.1	36.4	36.8	43.8	43.4	43.5	49.4	49.1	49.0
	R.	50.8	52.1	52.7	76.7	77.0	77.6	81.7	81.9	82.2	84.0	83.9	84.3	85.9	85.9	86.0
	S.	33.1	33.9	34.8	56.2	56.6	57.2	62.9	62.9	63.7	67.8	67.7	68.1	72.5	72.3	72.6

| a realistic painting of a large blue aircraft carrier | an impressionistic mosaic art of a small black backpack | a realistic photograph of a large black car | a realistic sketch of a large black mug | an impressionistic painting of a normal sized green train |

Fig. 1: Examples of synthetic images created with SDv2.1 and associated prompts.

v2-1 Model Card available on Hugging Face[1]. For each class across the four datasets, we produce 480 images using our synthetic template prompts as input for the stable diffusion model. All generated images are of 512×512 resolution. Examples of these synthetic images alongside their corresponding text prompts are displayed in Fig. 1.

B.2   LLM Prompts Generation

We utilize ChatGPT-3.5 [3] to create the LLM prompts employed in our comparative analysis of different prompt sources. Fig. 2 illustrates the process of prompting ChatGPT-3.5 to generate text prompts for specific class names. For each class, we produce 120 samples, and below are a few examples from the generated prompts:

– Bird:

---

[1] https://huggingface.co/stabilityai/stable-diffusion-2-1

Table 5: Domain-level results under adversarial attacks of ViT-B/16 on the datasets.

Dataset	Domains	FGSM						PGD-20						CW-20					
		ZS-C			1-shot			ZS-C			1-shot			ZS-C			1-shot		
		CLIP	Im.Aug	CLAP	CLIP	Im.Aug	CLAP	CLIP	Im.Aug	CLAP	CLIP	Im.Aug	CLAP	CLIP	Im.Aug	CLAP	CLIP	Im.Aug	CLAP
PACS	A.	76.3	79.3	79.3	61.2	78.0	87.3	1.7	2.2	1.8	16.0	42.1	63.1	1.5	2.0	2.3	0.5	1.1	1.7
	C.	94.9	95.0	94.0	66.5	84.2	95.1	33.3	37.7	35.6	33.3	57.2	86.1	28.8	34.0	33.2	11.9	23.6	31.8
	P.	91.6	90.3	91.7	67.4	80.8	92.1	5.7	7.0	6.7	27.1	55.0	69.8	4.7	4.9	5.8	0.7	2.7	4.1
	S.	84.5	87.5	89.8	71.6	74.6	83.8	75.8	78.4	79.2	63.0	66.3	74.6	74.5	76.8	77.9	62.7	65.4	70.3
VLCS	C.	55.3	53.8	55.5	25.8	28.8	25.3	4.4	5.1	4.7	2.0	5.2	2.5	2.9	3.1	3.5	0.7	1.2	1.0
	L.	49.4	45.5	50.6	27.0	32.6	30.4	15.2	14.9	16.0	6.4	8.9	8.0	12.4	11.2	13.0	6.1	8.3	7.7
	S.	61.7	58.1	62.5	48.0	46.9	51.6	13.2	13.9	14.0	8.6	10.7	10.0	9.2	8.8	10.2	8.3	7.9	8.4
	V.	65.3	63.2	65.6	36.5	40.1	41.0	7.5	7.9	7.9	5.3	9.4	8.9	5.2	4.8	5.6	2.9	2.8	2.9
OfficeHome	A.	55.3	53.8	55.5	25.8	28.8	25.3	4.4	5.1	4.7	2.0	5.2	2.5	2.9	3.1	3.5	0.7	1.2	1.0
	C.	49.4	45.5	50.6	27.0	32.6	30.4	15.2	14.9	16.0	6.4	8.9	8.0	12.4	11.2	13.0	6.1	8.3	7.7
	P.	61.7	58.1	62.5	48.0	46.9	51.6	13.2	13.9	14.0	8.6	10.7	10.0	9.2	8.8	10.2	8.3	7.9	8.4
	R.	65.3	63.2	65.6	36.5	40.1	41.0	7.5	7.9	7.9	5.3	9.4	8.9	5.2	4.8	5.6	2.9	2.8	2.9
DomainNet	C.	57.8	50.9	58.8	33.3	34.3	35.0	21.6	18.7	22.8	18.4	19.6	20.0	15.8	12.5	16.6	7.0	7.5	7.8
	I.	35.8	28.0	37.0	12.2	13.3	13.2	6.1	3.7	6.7	4.6	5.3	5.1	3.3	1.9	3.7	0.9	0.9	0.9
	P.	43.9	39.0	44.3	18.4	20.6	20.3	3.1	2.8	3.3	5.4	10.4	9.9	1.8	1.3	1.9	0.3	0.3	0.3
	Q.	12.9	10.3	13.2	10.9	10.8	11.1	8.4	6.8	8.6	5.4	5.4	5.6	7.1	5.4	7.4	4.9	4.8	5.1
	R.	62.1	55.9	62.4	34.5	35.9	36.5	7.1	6.5	7.5	17.6	19.7	19.6	4.5	3.4	4.7	1.2	1.4	1.4
	S.	49.1	43.3	49.7	25.7	26.0	27.5	17.8	15.5	18.6	13.6	14.4	15.1	13.4	10.2	13.9	5.0	5.2	5.6

- A pair of vibrant macaws converse in a lush, tropical rainforest, depicted in a lively, exotic wildlife painting.
- A solitary eagle watches over a vast, rugged canyon at sunrise, portrayed in a majestic, wilderness landscape photograph.

– Dog:

- A sleek Whippet races in a competitive dog track, illustrated in a fast-paced, dynamic sports style.
- A sturdy and reliable English Bulldog watching over a small shop, its solid presence reassuring to the owner.

– Car:

- A quirky art car parades through the streets in a colorful festival, captured in a fun, expressive style illustration.
- A high-tech, autonomous car maneuvers through a smart city environment, portrayed in a futuristic, sci-fi digital art piece.

– Chair:

- A folding chair at an outdoor wedding, elegantly decorated and part of a beautiful ceremony.
- A high-end executive chair in a law firm, projecting authority and professionalism.

– Person:

- An energetic coach motivates a team on a sports field, illustrated in an inspiring, leadership-focused painting.
- A graceful figure skater glides across an ice rink, captured in a delicate, winter-themed pastel drawing.

Table 6: Zero-Shot Performance on VLCS Dataset Across Varied Augmentation Combinations and Prompt Sources: ① Random Object Size Deletion, ② Random Object Color Deletion, ③ Random Image Type Deletion, ④ Random Art Style Deletion, ⑤ Random Swapping Order, ⑥ Addition of Gaussian Noise.

Method	Domains	CLIP (base)	①②③ ④⑤⑥	①②③ ④⑤	①②③ ④⑥	①②③ ④	③④⑤ ⑥	①②⑤ ⑥	EDA LLM	Rand.	Pr.St.	Temp.
	C.	99.7	99.9	99.8	99.9	99.8	99.9	99.9	97.9	99.7	99.9	99.9
ZS(C)	L.	61.8	66.6	62.3	67.0	62.2	66.2	67.7	66.2	69.0	67.3	66.5
	S.	70.1	78.1	75.5	78.0	74.3	78.5	78.0	73.2	76.9	73.5	76.9
	V.	73.9	82.8	80.6	83.2	79.3	82.7	84.9	72.6	81.8	81.8	81.9
	C.	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.8	99.9	99.9	99.9
ZS(CP)	L.	69.9	69.3	67.9	69.6	68.4	70.0	70.4	69.3	70.4	71.2	69.7
	S.	73.3	77.6	76.4	76.7	75.9	78.8	77.2	76.2	75.2	75.1	78.0
	V.	84.8	85.3	84.0	85.3	84.2	85.1	86.0	77.0	84.2	86.0	84.6
	C.	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9
ZS(PC)	L.	70.2	70.0	68.0	70.1	68.5	70.0	70.7	67.5	70.6	71.8	70.0
	S.	73.6	76.6	75.6	76.0	74.8	77.8	76.9	76.9	75.1	74.9	78.2
	V.	86.1	85.7	84.7	85.7	84.5	85.5	86.2	78.2	84.6	86.8	84.8
	C.	87.0	99.8	99.6	99.8	99.4	99.7	99.9	95.3	98.6	99.6	99.8
ZS(NC)	L.	55.9	65.2	61.3	65.6	60.5	65.4	65.9	63.0	66.7	64.0	64.7
	S.	61.4	75.6	70.3	75.2	68.3	74.9	75.3	68.9	73.3	69.8	73.0
	V.	68.9	80.1	75.2	80.4	73.8	79.4	82.9	69.3	79.6	77.2	78.6

C   Experiments on Other CLIP Model Scales

C.1   Experiments on ViT-L/14

We refined the output dimension to align with the input dimension of 768. The chosen latent dimensions were 448 and 640 for PACS and VLCS, respectively, and 768 for both OfficeHome and DomainNet. The inference weighting α was set to 0.1 for PACS, 0.03 for VLCS, 0.14 for OfficeHome, and 0.2 for DomainNet. All other training configurations remained consistent with the ViT-B/16 experiments across each dataset. The training configuration for Im.Aug was set the same as CLAP for each dataset, with the inference weighting α being 0.1 for PACS and 0.03 for the other three datasets.

Table 8 showcases the zero-shot results for the ViT-L/14 model using four distinct prompts, following the protocol established for the ViT-B/16 experiments. These results demonstrate that CLAP is more efficient than Im.Aug in enhancing zero-shot performance. Moreover, Tab. 9 illustrates that CLAP significantly reduces variations in zero-shot performance across different prompts, thereby confirming CLAP's performance improvements over CLIP across a range of model sizes. Detailed domain-level results are presented in Tab. 10, offering an in-depth analysis.

C.2   Experiments on ResNet50x16

To validate our approach on different model structures, we repeated zero-shot experiments on the ResNet50x16 model pre-trained with CLIP. Since the output

Table 7: Ablative study of hyperparameters on VLCS dataset using ViT-B/16 model.

Hyper-parameters	Value	Avg. top-1 acc. (%) (↑) using ViT-B/16 on VLCS dataset											
		ZS (C)				ZS (CP)				ZS (PC)			
		C.	L.	S.	V.	C.	L.	S.	V.	C.	L.	S.	V.
τ	0.1	99.9	67.6	77.5	84.2	99.9	70.9	74.9	85.9	99.9	71.2	74.6	86.3
	0.3	99.9	66.3	77.2	82.4	99.9	69.9	76.7	85.2	99.9	69.9	76.4	85.4
	0.5	99.9	67.7	78.0	84.9	99.9	70.4	77.2	86.0	99.9	70.7	76.9	86.2
	0.7	99.9	65.9	77.7	83.1	99.9	68.9	77.9	84.9	99.9	69.6	77.7	85.0
	0.9	99.9	66.0	77.6	83.3	99.9	69.0	77.9	85.0	99.9	69.7	77.5	85.0
Lantent dim.	128.0	99.9	66.0	77.6	82.6	99.9	70.0	77.4	85.4	99.9	70.1	77.1	85.7
	192.0	99.9	64.9	77.9	83.0	99.9	68.9	78.0	85.6	99.9	69.0	77.8	86.0
	256.0	99.9	63.8	77.6	82.7	99.9	67.6	78.7	84.8	99.9	67.8	78.6	85.2
	320.0	99.9	66.0	77.8	82.9	99.9	69.2	78.1	85.3	99.9	69.7	77.7	85.5
	384.0	99.9	65.8	76.9	82.8	99.9	69.4	77.5	85.3	99.9	69.6	77.0	85.5
	448.0	99.9	65.8	77.4	82.1	99.9	69.7	77.6	84.9	99.9	69.9	77.1	85.6
	512.0	99.9	67.7	78.0	84.9	99.9	70.4	77.2	86.0	99.9	70.7	76.9	86.2
α	$10^{-1.5}$	99.9	66.5	77.9	83.1	99.9	70.4	77.1	86.0	99.9	70.3	76.6	86.1
	10^{-1}	99.9	69.5	77.5	85.7	99.9	70.4	77.1	86.2	99.9	70.9	76.5	86.1
	$10^{-0.5}$	99.9	70.6	75.2	85.5	99.9	70.7	75.7	85.9	99.9	71.0	75.1	85.7
	10^{0}	99.8	71.5	73.5	83.5	99.9	71.7	74.4	85.8	99.8	72.3	73.5	85.5
	$10^{0.5}$	99.8	72.0	73.1	85.5	99.8	72.2	73.7	85.7	99.8	72.5	72.9	85.6
	10^{1}	99.8	72.1	72.8	85.4	99.8	72.3	73.4	85.7	99.8	72.5	72.9	85.5
	$10^{1.5}$	99.8	72.1	72.8	85.4	99.8	72.2	73.3	85.7	99.8	72.6	72.7	85.5

dimension of CLIP is the same as ViT-B/16, we used the same training configuration as ViT-B/16 for training Im.Aug and CLAP. For inference, we refined the weighting coefficient α to 0.1, 1, 0.03, and 0.1 for Im.Aug, and 0.03, 0.2, 0.06, and 0.1 for CLAP, for PACS, VLCS, OfficeHome, and DomainNet respectively.

Table 11 showcases the zero-shot results for ResNet50x16 model across different prompts, substantiating that CLAP is more effective than Im.Aug in refining CLIP features. Moreover, Tab. 12 illustrates that both Im.Aug and CLAP reduce variations in zero-shot performance across different prompts, with the improvement of CLAP being more significant. The results validate our approach across different model scales, including both ViT-based and CNN-based structures. Domain-level results are detailed in Tab. 13.

D   Discussion

D.1   Rationale behind CLAP's Foundation on CLIP

The primary challenge in cross-modal transferability lies in the significant domain gap between text and image data, which typically hinders the direct application of models trained in one modality to another. For a causal explaination, despite the consistency of the content variable that dictates the object label across modalities, the generative processes from latent variables to observations inherent to each modality differ markedly. The CLIP model, trained on a comprehensive dataset of image-text pairs with a symmetric InfoNCE loss, significantly ameliorates this issue. By aligning the features of text and images into similar patterns, it facilitates leveraging a network trained atop the CLIP encoder of
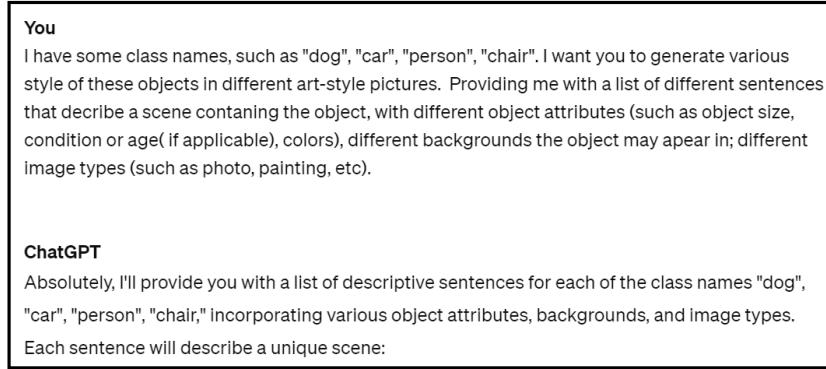
Fig. 2: The prompting method we use for generating text prompts with ChatGPT-3.5.

Table 8: Zero-shot performance across four prompts ("C", "PC", "CP") and 1 noised prompts ("NC") with CLIP pre-trained ViT-L/14 model. CLAP demonstrates consistent gains in zero-shot performance across all datasets, validating its effectiveness.

Prompt	Method	Zero-shot performance, avg. top-1 acc. (%) (↑)				
		PACS	VLCS	OfficeHome	DomainNet	Overall
ZS(C)	CLIP	97.6	77.1	85.9	63.2	80.9
	Im.Aug	98.3	78.5	86.0	63.4	81.6
	CLAP	**98.5**	**80.7**	**87.5**	**64.2**	**82.7**
ZS(CP)	CLIP	97.3	80.6	86.0	62.0	81.5
	Im.Aug	98.3	81.1	86.1	62.4	82.0
	CLAP	**98.5**	**81.4**	**87.9**	**63.7**	**82.9**
ZS(PC)	CLIP	98.4	81.7	86.5	63.5	82.5
	Im.Aug	**98.6**	81.9	86.6	63.7	82.7
	CLAP	**98.6**	**82.2**	**88.0**	**64.5**	**83.3**
ZS(NC)	CLIP	91.0	65.5	77.1	55.4	72.3
	Im.Aug	95.6	69.3	77.1	55.7	74.4
	CLAP	**98.5**	**73.1**	**81.3**	**58.3**	**77.8**

one modality as a viable proxy for the other. Consequently, this allows for the direct application of the disentangled network trained in the text modality atop CLIP's image encoder to refine representations.

D.2   Impact of Image and Text Augmentations

Identifying pure content factors poses a significant challenge. This difficulty primarily arises from the need for finding effective augmentations of observational data to alter style factors significantly while preserving content integrity.

Through the cross-modal alignment provided by CLIP, we discovered that disentangling in one modality can seamlessly improve representations in both modalities. The impact of image augmentations has been well-explored and found effective at preserving content, but traditional methods do not impose sufficient changes to remove all style information. Our exploration of text augmentations

Table 9: CLAP reduces the variance in zero-shot performance across different prompts with CLIP pre-trained ViT-L/14 model.

Metric	Method	Zero-shot variance, avg. top-1 acc. (%) ($\downarrow$)				
		PACS	VLCS	OfficeHome	DomainNet	Overall
	CLIP	1.0	4.6	0.6	1.5	1.9
R	Im.Aug	0.3	3.4	0.6	1.3	1.4
	CLAP	**0.1**	**1.5**	**0.4**	**0.7**	**0.7**
	CLIP	0.4	2.0	0.3	0.6	0.8
δ	Im.Aug	0.1	1.5	0.3	0.5	0.6
	CLAP	**0.0**	**0.6**	**0.2**	**0.3**	**0.3**
	CLIP	6.6	11.5	8.8	7.8	8.7
$\Delta_{(NC)}$	Im.Aug	2.7	9.2	8.9	7.7	7.1
	CLAP	**0.1**	**7.7**	**6.3**	**5.9**	**5.0**

Table 10: Domain-level zero-shot results of the ViT-L/14 model on the test datasets.

Datasets	Domains	Domain-level avg. top-1 acc. (%) of zero-shot performance using ViT-L/14 ($\uparrow$)											
		ZS(C)			ZS(CP)			ZS(PC)			ZS(NC)		
		CLIP	Im.Aug	CLAP	CLIP	Im.Aug	CLAP	CLIP	Im.Aug	CLAP	CLIP	Im.Aug	CLAP
PACS	A.	97.2	98.0	98.8	96.8	98.0	98.5	98.7	98.8	98.9	85.6	91.6	98.6
	C.	99.5	99.6	99.8	98.3	99.6	99.7	99.5	99.6	99.7	95.9	98.1	99.6
	P.	99.9	100.0	100.0	99.4	99.5	100.0	99.9	100.0	99.9	91.1	97.5	99.9
	S.	93.8	95.7	95.5	94.8	96.0	95.7	95.4	95.9	95.8	91.5	95.2	95.8
VLCS	C.	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	87.5	87.9	94.4
	L.	57.4	60.1	64.3	71.3	71.6	72.6	71.7	72.0	72.6	53.8	59.7	60.7
	S.	71.0	72.4	74.4	66.2	67.4	66.8	69.9	70.4	69.9	55.9	60.5	62.9
	V.	80.0	81.6	84.3	85.2	85.7	86.2	85.1	85.3	86.4	65.0	69.3	74.3
OfficeHome	A.	86.2	86.3	87.7	85.7	86.2	88.1	87.0	87.0	87.8	78.1	77.1	80.7
	C.	73.3	73.4	75.7	73.8	73.4	76.0	73.1	73.5	76.0	65.9	66.3	70.6
	P.	92.0	91.8	93.6	92.3	92.4	94.3	92.9	92.8	94.1	80.7	81.0	86.8
	R.	92.2	92.7	93.0	92.2	92.4	93.4	93.1	93.3	93.9	83.8	84.0	86.9
DomainNet	C.	78.4	78.5	79.1	77.5	77.7	78.8	79.4	79.4	79.7	70.0	70.4	72.8
	I.	52.9	53.0	54.6	50.4	50.7	53.6	51.7	52.0	53.9	45.3	45.2	48.8
	P.	70.4	70.8	72.4	68.9	69.9	72.1	69.9	70.6	72.7	59.9	60.3	64.8
	Q.	21.5	21.6	22.5	20.6	20.9	21.7	22.6	22.8	22.9	17.9	18.4	20.2
	R.	85.8	85.9	85.9	85.3	85.5	85.7	86.3	86.4	86.2	77.5	77.5	78.7
	S.	70.2	70.4	70.7	69.4	69.8	70.6	71.0	71.3	71.5	62.0	62.2	64.6

reveals that the logical structure of text and the relative ease of implementing style changes can have a significant impact on achieving disentanglement. However, more efficient methods are worthy of exploration.

A promising direction for future research is to explore efficient combinations of both modalities to enhance disentangled semantics. As each modality has its unique advantages—Text data recapitulates properties well since it is preprocessed by human intelligence, while image data is more precise in depicting the exact same objects or events due to its more detailed nature—the impact of combining augmentations of both modalities could be substantial.

Table 11: Zero-shot performance with CLIP pre-trained ResNet50x16 model. CLAP demonstrates consistent enhancement across all datasets, validating its effectiveness.

Prompt	Method	Zero-shot performance, avg. top-1 acc. (%) (↑)				
		PACS	VLCS	OfficeHome	DomainNet	Overall
ZS(C)	CLIP	96.1	70.4	80.4	57.1	76.0
	Im.Aug	96.4	74.7	80.4	57.1	77.2
	CLAP	**97.0**	**79.9**	**81.6**	**58.0**	**79.1**
ZS(CP)	CLIP	95.0	73.5	79.0	56.1	75.9
	Im.Aug	95.7	75.8	79.3	56.5	76.8
	CLAP	**96.7**	**80.3**	**79.9**	**57.4**	**78.6**
ZS(PC)	CLIP	96.5	78.4	81.7	57.1	78.4
	Im.Aug	97.0	79.8	81.8	57.4	79.0
	CLAP	**96.8**	**80.1**	**82.5**	**58.2**	**79.4**
ZS(NC)	CLIP	86.4	61.2	69.3	48.2	66.3
	Im.Aug	88.3	71.3	69.5	48.7	69.4
	CLAP	**94.9**	**80.1**	**71.9**	**50.6**	**74.4**

Table 12: CLAP consistently reduces variances in zero-shot performance across different prompts with CLIP pre-trained ResNet50x16 model, validating its effectiveness.

Metric	Method	Zero-shot variance, avg. top-1 acc. (%) (↓)				
		PACS	VLCS	OfficeHome	DomainNet	Overall
R	CLIP	1.5	8.0	2.7	1.1	3.3
	Im.Aug	1.3	5.1	**2.5**	0.9	2.4
	CLAP	**0.3**	**0.4**	2.6	**0.8**	**1.0**
δ	CLIP	0.6	3.3	1.1	0.5	1.4
	Im.Aug	0.5	2.2	**1.0**	0.4	1.0
	CLAP	**0.1**	**0.2**	1.1	**0.3**	**0.4**
$\Delta_{(NC)}$	CLIP	9.7	9.3	11.1	8.9	9.7
	Im.Aug	8.1	3.5	10.9	8.5	7.7
	CLAP	**2.1**	**-0.1**	**9.7**	**7.5**	**4.8**

Table 13: Domain-level zero-shot results using RestNet50x16 on the test datasets.

Datasets	Domains	Domain-level avg. top-1 acc. (%) of zero-shot performance using RN50x16 (↑)											
		ZS(C)			ZS(CP)			ZS(PC)			ZS(NC)		
		CLIP	Im.Aug	CLAP	CLIP	Im.Aug	CLAP	CLIP	Im.Aug	CLAP	CLIP	Im.Aug	CLAP
PACS	A.	95.7	95.8	97.2	93.7	95.5	96.5	95.7	96.7	96.8	81.2	84.0	94.5
	C.	98.3	98.2	99.0	98.1	98.8	99.0	98.6	98.7	98.9	92.3	93.2	98.0
	P.	98.9	98.6	99.9	98.4	97.8	99.8	99.8	99.9	99.9	85.3	87.3	95.2
	S.	91.5	93.1	91.9	89.7	90.8	91.5	91.8	92.9	91.6	86.9	88.6	92.1
VLCS	C.	96.8	97.1	99.3	99.7	99.4	99.3	99.7	99.6	99.4	75.6	89.3	99.4
	L.	53.4	60.8	65.9	51.6	58.9	67.3	59.5	68.1	66.8	54.1	60.6	67.0
	S.	63.2	70.9	69.5	68.0	72.9	69.5	72.9	73.7	69.0	52.0	66.7	69.5
	V.	68.4	70.1	85.2	74.5	72.1	85.3	81.7	78.0	85.2	63.1	68.5	84.5
OfficeHome	A.	82.2	82.5	83.5	79.7	79.9	80.6	82.0	82.4	83.4	67.7	68.9	72.1
	C.	63.0	62.9	64.7	61.7	62.2	62.8	65.4	65.3	66.1	54.6	55.0	56.8
	P.	88.2	87.9	89.0	87.4	87.5	88.5	90.0	89.9	90.6	75.4	75.3	78.2
	R.	88.1	88.2	89.1	87.3	87.5	87.6	89.2	89.5	89.7	79.5	78.9	80.3
DomainNet	C.	69.0	68.9	69.6	68.6	68.6	69.4	69.9	70.0	70.4	59.5	60.1	61.4
	I.	51.0	51.1	52.7	48.2	49.0	50.6	48.2	48.9	50.7	41.2	41.6	44.3
	P.	65.2	65.6	66.5	63.7	64.4	65.6	65.4	65.9	67.0	53.5	54.3	56.8
	Q.	11.8	11.9	12.7	12.3	12.6	13.1	11.8	12.2	12.7	9.3	9.7	11.0
	R.	82.1	82.2	83.1	81.6	81.8	82.6	83.3	83.4	83.8	72.9	73.0	74.7
	S.	63.2	63.1	63.6	62.0	62.4	63.4	63.9	63.8	64.6	53.1	53.4	55.3