A Language Anchor-Guided Method for Robust Noisy Domain Generalization

Anonymous authors
Paper under double-blind review

Abstract

Real-world machine learning applications are often hindered by two critical challenges: distribution shift and label noise. Networks inherently tend to overfit to redundant, uninformative features present in the training distribution, which undermines their ability to generalize effectively to the target domain's distribution. The presence of noisy data further exacerbates this issue by inducing additional overfitting to noise, causing existing domain generalization methods to fail in effectively distinguishing invariant features from spurious ones. To address these challenges, we propose **A**nchor **A**lignment and **A**daptive **W**eighting (A^3W), a novel algorithm based on sample reweighting guided by natural language processing (NLP) anchors that seeks to extract representative features. In particular, A^3W leverages semantic representations derived from natural language models to serve as a source of domain-invariant prior knowledge. We also introduce a weighted loss function that dynamically adjusts the contribution of each sample based on its distance to the corresponding NLP anchor, thereby improving the model's resilience to noisy labels. Extensive experiments on benchmark datasets demonstrate that A^3W outperforms state-of-the-art domain generalization methods, yielding significant improvements in both accuracy and robustness across various datasets and noise levels.

1 Introduction

Domain Generalization (DG) has emerged as a pivotal algorithm in machine learning, aiming to develop models that can maintain high performance on previously unseen environments—or *domains*. Traditional methods often assume that training and test data share the same distribution, yet in real-world scenarios, there is frequently a substantial shift between these distributions. This phenomenon, widely referred to as *domain shift*, can cause severe performance degradation in tasks spanning computer vision, natural language processing, and medical image analysis (Wang et al., 2022). As shown in Figure 1(a)(b), even within the same class label, the distribution of feature representations can vary considerably. This variation may stem from differences in image acquisition conditions—such as lighting variations, changes in pose, or complex background environments—and even from more subtle domain-specific factors like sensor noise or camera calibration differences. Such intra-class variability poses a significant challenge for developing accurate and adaptable models, which must learn to extract invariant features that capture the true semantic essence of the class while ignoring irrelevant variations. In response, DG task has been proposed to enable robust model behavior without further fine-tuning, thereby facilitating broader applicability in diverse domains (Muandet et al., 2013; Li et al., 2018). Over time, researchers have explored various ways to align or unify source and target domains, including strategies that learn *domain-invariant* features through multi-task autoencoders, augmented architecture, or adversarial alignment (Ghifary et al., 2015; Ganin & Lempitsky, 2015; Volpi et al., 2018).

One persistent obstacle in DG is the prevalence of *spurious correlations* in deep neural networks, where models inadvertently link target labels to irrelevant, domain-specific features (Arjovsky et al., 2019; Qiao & Low, 2024). For instance, a classifier tasked with recognizing animals might rely on particular backgrounds for prediction, causing it to misclassify animals photographed in unfamiliar settings (Beery et al., 2018).

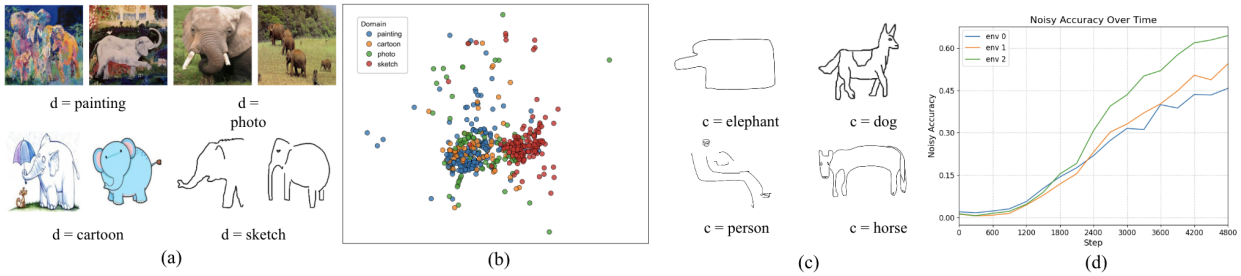

Figure 1: Illustration of domain shift using four distinct domains—painting, photo, cartoon, and sketch—for the same object class. In (a), the visual appearance of "elephant" varies substantially across domains, underscoring significant style discrepancies. In (b), the t-SNE projection shows that even for the same class, the distribution of features differs across domains, highlighting the inherent challenges of domain generalization. In (c), unclear or mislabeled samples introduces additional noise, further exacerbating the difficulty of achieving robust generalization. In (d), we show the accuracy of a network trained on noisy data: ideally, the model should resist overfitting to noise, but the graph indicates a steady increase in noise accuracy over time, suggesting progressive overfitting to noisy labels.

Such spurious correlations frequently arise because modern networks have the capacity to overfit even low-level noise, thereby anchoring their decisions on non-causal cues (Rahman et al., 2024). To address these pitfalls, researchers have proposed a variety of techniques, from causal inference algorithms like Invariant Risk Minimization (IRM) (Arjovsky et al., 2019) to methods specifically designed to disentangle spurious and invariant features (Wang et al., 2024). For example, Wu et al. (2023) introduced the Discover and Cure (DISC) strategy, which detects suspicious features and systematically mitigates their impact by leveraging interpretable domain knowledge. The development of methods rooted in causal representation learning also arises to mitigate the reliance on spurious correlations (Lv et al., 2022). Despite these advances, achieving robust generalization under significant domain shift and considerable label noise persists as an open research challenge that necessitates more comprehensive methods and better theoretical insights.

Existing domain generalization (DG) approaches have predominantly focused on aligning feature distributions or disentangling spurious factors using data-driven methods (Wang et al., 2022). While these techniques have paved the way for significant advances, they often encounter considerable difficulties when label noise is prevalent or when data lack distinct domain-invariant cues (Nigam et al., 2020), leading to rapid degradation in network performance as noise intensity increases. This challenge is further exacerbated in large, overparameterized networks that are prone to memorizing spurious correlations, thereby amplifying errors. Indeed, estimates suggest that 8% to 38.5% of real-world data may suffer from label corruption, undermining the generalizability of models that fail to distinguish genuine signals from incidental patterns (Song et al., 2022a). Although some DG algorithms inherently exhibit robustness to label noise, in general cases they lack performance superiority compared to empirical risk minimization (ERM) baseline (Qiao & Low, 2024). Consequently, understanding how to boost robustness against both domain shifts and noisy labels remains a pressing goal in building truly reliable artificial intelligence systems.

In light of these challenges, we consider an alternative approach that focuses on introducing novel sources of features from diverse perspectives to guide the learning pipeline. An emerging line of research has investigated the role of *external knowledge*, which can guide models toward more semantically grounded representations (Dash et al., 2022). Some studies incorporate domain knowledge directly into the learning pipeline, for instance by embedding specialized scientific insights into feature extraction or loss design (Von Rueden et al., 2021). In other cases, researchers have shown that domain knowledge can significantly enhance model interpretability and robustness—for example, in laser-induced breakdown spectroscopy, where integrated expert knowledge improved quantification performance (Song et al., 2022b). Another promising direction involves multi-modal alignment, in which semantic text descriptions serve as stable anchors that help networks transcend purely visual or sensor-driven biases (Liu & Wang, 2023). Models like CLIP (Radford et al., 2021) exemplify this principle by projecting both text and images into a shared embedding space, improving generalization across dissimilar environments. Yet critical barriers remain regarding how to ensure that the

guidance remains valid across diverse domains and how to structure knowledge encoding effectively to align with modern deep learning architectures (Dash et al., 2022).

Inspired by this recent trend of knowledge integration, we propose a new algorithm called **A**nchor **A**lignment and **A**daptive **W**eighting (A^3W), which integrates external knowledge—specifically, linguistic cues derived from large-scale language models—into the DG pipeline. Our central insight is that *NLP anchors*, such as text embeddings from CLIP or analogous language models, provide *domain-invariant* references capable of steering learned representations away from spurious and noisy cues. By aligning intermediate feature spaces with these carefully chosen textual anchors, A^3W encourages semantic consistency while limiting the influence of mislabeled or outlier samples. Moreover, we introduce a *weighted loss function* that dynamically modulates each sample's impact based on its proximity to an anchor, thereby enhancing robustness to label noise. We validate our method through extensive experiments on multiple DG benchmarks, confirming that A^3W surpasses existing approaches in terms of both accuracy and resilience to shifting and noisy data. Our contributions can be summarized as follows:

- We propose an iterative update algorithm that unifies external semantic knowledge and image-based features, enabling more robust and interpretable generalization across unseen domains.
- We introduce the concept of NLP anchors derived from large-scale language models (e.g., CLIP), which provide domain-invariant and semantically rich feature constraints that significantly improve model robustness. Furthermore, we propose a novel *weighted loss function* that dynamically adjusts each sample's contribution based on its distance to the corresponding NLP anchor. By assigning higher importance to samples that are closer to the learned semantic representations and lower importance to outliers, our approach effectively enhances feature alignment, ultimately leading to improved generalization across unseen domains.
- We conduct extensive experiments on multiple domain generalization benchmarks to demonstrate that A^3W outperforms state-of-the-art methods in terms of accuracy, robustness, and adaptability.

2 Related Work

2.1 Domain Generalization

Domain generalization (DG) addresses the challenge of training models that can perform well on previously unseen domains, without access to labeled examples from those domains at training time. A significant challenge in DG is the presence of *spurious correlations* in deep neural networks, which hinders generalization across diverse settings (Arjovsky et al., 2019). These correlations arise when models inadvertently rely on domain-specific artifacts—features that are incidentally correlated with the target labels in the training data—rather than on the truly invariant properties that are essential for robust performance. Consequently, when models encounter data from an unseen domain, their over-reliance on these spurious cues leads to performance degradation when deployed in unseen domains. Early methods focused on learning domain-invariant representations through shallow or deep feature alignments, aiming to eliminate domain-specific information while preserving task-relevant features. For instance, Muandet et al. (2013) propose Domain-Invariant Component Analysis (DICA), a kernel-based method to learn invariant features by minimizing the discrepancy across source domains. Ghifary et al. (2015) use multi-task autoencoders to learn generic representations.

More recent works resort to data augmentation or adversarial strategies to synthesize novel training distributions, thereby exposing models to a richer variety of samples. Volpi et al. (2018) introduced adversarial data augmentation by generating worst-case perturbations to expose the model to a broader set of variations during training, thereby improving its ability to handle unseen shifts. Ganin & Lempitsky (2015) introduced domain-adversarial training using gradient reversal layers to align feature distributions. Li et al. (2018) explored meta-learning strategies to enhance DG by simulating domain shifts during training. Shankar et al. (2018) proposed a generalization method using domain-specific perturbations to improve robustness. Another class of augmentation-based methods applies Mixup strategies, interpolating data points across domains to improve robustness (Yan et al., 2020). Style transfer methods have also been introduced to

augment training data with diverse visual styles, improving out-of-distribution performance (Nam et al., 2021). (Carlucci et al., 2019) proposed a self-supervised approach that solves jigsaw puzzles to learn more generalized features.

Beyond augmentation, several methods have turned their focus to regularization to improve generalization. Balaji et al. (2018) introduced a regularization algorithm by learning a regularizer modeling the objective of DG that a model trained on one domain to generalize effectively. Dou et al. (2019) utilized episodic training to simulate domain shifts and improve generalization. Zhou et al. (2021) introduced a regularization approach based on domain-adaptive ensemble learning to enhance a model's ability to generalize across different domains. Furthermore, recent research has revisited the issue of spurious correlations in DG, proposing methods to mitigate their impact. Qin et al. (2024) explored building a structural causal model for representation learning to address spurious correlations. Ma et al. (2024) introduced FedCD, a federated domain generalization algorithm that employs a spurious correlation intervener for self-supervised feature intervention and a risk extrapolation aggregation strategy to reduce reliance on misleading shortcuts and boost performance on unseen domains. These studies underscore the importance of addressing spurious correlations to enhance DG.

2.2 Learning under noisy labels

Learning under noisy labels addresses the challenge of training models when annotated data contain errors or inconsistencies. Such noise often arises in real-world scenarios due to human annotation mistakes, weak labeling processes, or automated data collection pipelines. When training with noisy labels, deep neural networks can overfit to incorrect annotations, resulting in degraded performance and reduced robustness (Song et al., 2022a). Initial methods focused on designing robust loss functions and label-correction strategies to mitigate the impact of label noise. For example, Reed et al. (2014) introduced a "bootstrapping" approach that combines model predictions with the (potentially noisy) labels to guide learning. Goldberger & Ben-Reuven (2017) proposed adding a noise adaptation layer to model the corruption process directly. Veit et al. (2017) leveraged a small clean dataset to estimate noise statistics and correct the loss for mislabeled examples. These methods laid the groundwork for more advanced noisy-label handling techniques.

Subsequent research explored "co-teaching" strategies to further reduce the impact of noisy annotations. Han et al. (2018) proposed a co-teaching algorithm wherein two networks train simultaneously, exchanging likely clean samples to avoid memorizing label noise. MentorNet (Jiang et al., 2018) introduced a curriculum-based approach, using a "mentor" network to select reliable samples for the "student" network, progressively ignoring data points suspected to be noisy. A complementary direction has focused on iterative label refinement. Tanaka et al. (2018) presented a joint optimization algorithm that updates labels alongside network parameters, gradually reducing noise in the dataset. To improve robustness further, recent efforts incorporate data augmentation strategies specifically tailored for noisy labels. For instance, Nishi et al. (2021) systematically studied augmentation techniques to enhance the network's tolerance to corrupted annotations.

Recent studies delve deeper into noisy-label learning in conjunction with large-scale or real-world datasets. DivideMix (Li et al., 2020) treated noisy-label learning as a semi-supervised problem, splitting data dynamically into clean and noisy sets for more targeted training. Song et al. (2022a) provided a comprehensive survey on deep learning with noisy labels, highlighting emerging trends such as instance-dependent noise modeling and robust early-learning regularization. These approaches emphasize balancing label correction, sample selection, and model regularization. Robust learning under noisy labels has become increasingly important for domain generalization tasks, as noisy annotations can exacerbate spurious correlations and hinder out-of-domain performance. Consequently, integrating advanced noisy-label handling into DG pipelines remains an active area of research. Some recent research has begun to address the intersection of label noise and domain shifts. Qiao & Low (2024) indicated that while label noise can sometimes serve as a form of regularization, in scenarios with significant domain shifts it may reinforce spurious correlations, ultimately degrading performance on unseen domains. These findings underscore the need for integrated approaches that jointly mitigate label noise and domain shifts to achieve robust out-of-distribution generalization.

3 Method

3.1 Problem Formulation

Let $\mathcal{D}_S$ denote a set of source domains, and $\mathcal{D}_T$ denote the target domain. We assume that $\mathcal{D}_S$ and $\mathcal{D}_T$ share the input space $\mathcal{X}$ and label space $\mathcal{Y}$, in which each domain $d \in \{1, \ldots, \mathcal{D}\}$ has data drawn from a joint distribution $P_d(X, Y)$ where $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$. In practice, the observed labels are corrupted by noise, i.e. for each sample with true label y, we observe a noisy label $\tilde{y}$ generated by:

$$\tilde{y} = \begin{cases} y, & \text{with probability } 1-p, \\ \tilde{y}' \sim Q(\cdot \mid y), & \text{with probability } p, \end{cases}$$

where p is the probability that the label is corrupted, and $Q(\tilde{y} \mid y)$ is a conditional distribution over possible noisy labels given y. For a classification problem with C classes with $c \in \{1, \ldots, C\}$, this means that the true label is flipped to a different class with probability p.

Noisy Domain Generalization: The proposed method aims to generalize to unseen domains while preventing overfitting to noisy data. In this setting, the training dataset $\mathcal{S}_{tr}$ is contaminated with label noise. We assume that the testing dataset $\mathcal{S}_{te}$ is drawn from a clean distribution with labels $\mathcal{Y}_{te}$. Our objective is to train a model on the noisy training dataset $\mathcal{S}_{tr}$ that learns robust and domain-invariant representations, thereby minimizing the classification error on the clean target domain $\mathcal{D}_{te}$.

3.2 Motivation

Domain generalization is a challenging problem primarily due to the tendency of networks to learn spurious correlations between features and labels. Such correlations can cause overfitting, even in tasks where the network achieves high performance on the training data. Observations from several works in DG highlighted the challenge for standard networks to learn representations that are robust or invariant enough to generalize well to unseen domains. The DomainNet (Peng et al., 2019) benchmark revealed that many current models struggle to transfer knowledge effectively across heterogeneous data sources. Gulrajani & Lopez-Paz (2020) showed that numerous DG algorithms fail to significantly outperform a simple empirical risk minimization (ERM) baseline. Salaudeen & Koyejo (2022) demonstrated that incomplete constraints can lead to suboptimal generalization performance because the network may not fully disentangle invariant from spurious features. On the other hand, Ben-David et al. (2010) in their work proved that while many methods can enforce domain invariance on the training domains, this invariance sometimes comes at the expense of losing the discriminative power of the features in unseen domains.

Natural occurrences of noise in datasets further exacerbated network overfitting. Noise can arise from a variety of sources, such as sensor inaccuracies, annotation errors, and environmental variations. Figure 1(c) presents examples of real-world noise. The figure displays sample images from the PACS (Li et al., 2017) dataset, where some images are either incorrectly annotated or are ambiguous and unclear even to the human eye. As of current DG algorithms, Qiao & Low (2024) has proved that DG algorithms inherently provide label-noise robustness; however, our observations indicate that performance still drops significantly as noise levels increase. Figure 1(d) illustrates the impact of noise on network fitting. Ideally, a network trained on noisy data would exhibit relatively stable accuracy throughout training iterations, indicating resistance to overfitting. However, the sharp increase in accuracy across all domains as training progresses suggests that the network is overfitting to the noise. This observation led us to rethink an alternative way to provide more guidance to the feature extraction process for learning better representation. Inspired by works on leveraging external knowledge for classification to bridge representation from different modalities, we propose to harness the power of semantic embeddings to provide systematic guidance during training. In our approach, we seek to mitigate the impact of data that exhibit significant noise or deviate from the primary training distribution. By incorporating semantic guidance, our aim is to weight samples based on their semantic consistency with predefined class prototypes or anchors. Formally, a semantic anchor $\mathbf{a_c} \in \mathbb{R}^m$ is computed for each class c, and the goal is to encourage the feature extractor $f : \mathcal{X} \to \mathbb{R}^m$ to map an input $\mathbf{x}$ to a representation $f(\mathbf{x})$ that aligns with the corresponding anchor $\mathbf{a_{\tilde{y}}}$ even in the presence of noise.

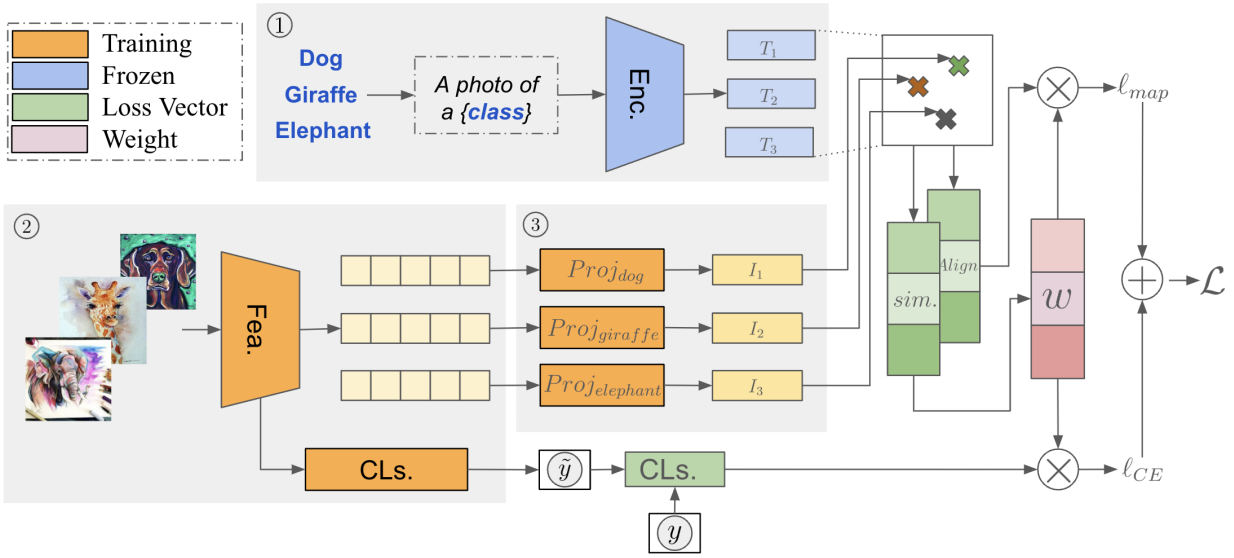

Figure 2: Architecture of A^3W. This diagram illustrates the end-to-end workflow and key components of A^3W. The encoder (Enc.) converts input text into embeddings, which are then refined by the featurizer (Fea.) into a more informative representation. The classifier (CLs.) leverages these refined features for prediction. Additionally, the similarity module (sim.) computes the cosine similarity between the embedding anchor and the projected features, while the alignment module (align.) creates deep copies of this similarity for weight (w) computation.

3.3 Algorithm for A^3W

To address the challenges above, the algorithm of A^3W is designed to learn a robust feature representation and classifier that generalizes well to the unseen target domain $\mathcal{D}_T$, despite the presence of noisy labels and distribution shifts across domains. At its core, A^3W aims to align the projected features extracted from input images with fixed semantic anchors that act as dependable sources of knowledge. Furthermore, rather than employing a binary selection mechanism which can discard valuable information, our approach uses a continuous weighting scheme that assigns each sample an importance value, effectively preserving the gradient contributions from all samples and enabling smoother, more adaptive optimization. The overall architecture is shown in Figure 2, where we first obtain the natural language processing (NLP) anchors, and then use them to iteratively guide the updates for the featurizer, classifier, and projection layers. These stages ensure that the model learns domain-invariant representations while maintaining stable optimization. The process mainly consists of three steps, with steps 2-3 being iterative:

1) NLP anchor setting phase: we start by computing semantic anchors using a pretrained language-vision model (i.e., CLIP) as text encoder. For each class c, a text prompt is generated using a template, which is tokenized and encoded by the CLIP model. The anchors $\tilde{\mathbf{a}}_\mathbf{c}$ are then stacked into an anchor matrix and are used to initialize a set of linear projectors $\{Proj_c\}_{c=1}^{C}$ that map image features into the same semantic space.

2) Main Model Update: we subsequently update the featurizer and classifier to improve classification performance using the combined loss of the weighted alignment and cross-entropy. During **inference**, the exponential moving average (EMA) network, initialized as a deep copy of the primary network, is employed to produce predictions, as it generally yields more stable and robust outputs compared to the primary network.

3) Mapping Layer Optimization: Each class is assigned a trainable mapping layer $Proj_c$, which projects feature embeddings into the NLP anchor space. The mapping layers are updated iteratively using our weighted loss function.

3.4 Obtaining NLP Anchors via CLIP Convention

In our method, the text encoder $T(\cdot)$ is based on a transformer architecture, as used in the CLIP model. For a given class c, we first construct a text prompt by appending a fixed template to the class name, and then encode this prompt using a pretrained text encoder $T(\cdot)$ to obtain an unnormalized anchor $\mathbf{a_c}$. To ensure that the anchor lies on the unit hypersphere, we normalize it, yielding $\tilde{\mathbf{a}}_\mathbf{c}$. The formulation is given by:

$$\begin{aligned} Prompt(c) &= \texttt{"a photo of a } c\texttt{"} \\ a_c &= T\big(Prompt(c)\big) \end{aligned} \tag{1}$$

The process begins by converting a given text prompt $Prompt(c)$ into a sequence of discrete tokens using a tokenizer. Let $t = [t_1, t_2, \ldots, t_L]$ denote the resulting token sequence, where L is the length of the sequence. Each token t_i is then mapped to a continuous vector via an embedding matrix $\mathbf{E} \in \mathbb{R}^{V \times m}$, where V is the vocabulary size and d is the embedding dimension:

$$\mathbf{e_i} = \mathbf{E}(t_i), \quad \text{for } i = 1, \ldots, L. \tag{2}$$

These token embeddings are augmented with positional encodings ($\mathbf{p}$) to incorporate the order of the tokens, resulting in a sequence of enriched embeddings:

$$\tilde{\mathbf{e}}_\mathbf{i} = \mathbf{e_i} + \mathbf{p_i}, \quad \text{for } i = 1, \ldots, L, \tag{3}$$

where p_i is the positional encoding for the i-th token. The sequence $\{\tilde{e}_1, \tilde{e}_2, \ldots, \tilde{e}_L\}$ is then fed into a transformer encoder, which consists of multiple layers of self-attention and feedforward networks. This produces a set of contextualized token representations:

$$\mathbf{H} = \text{Transformer}\big([\tilde{\mathbf{e}}_\mathbf{1}, \tilde{\mathbf{e}}_\mathbf{2}, \ldots, \tilde{\mathbf{e}}_\mathbf{L}]\big) = [\mathbf{h_1}, \mathbf{h_2}, \ldots, \mathbf{h_L}]. \tag{4}$$

Typically, a special token (such as $[EOS]$ or $[CLS]$) is appended to the input sequence, and its corresponding output h_c is used as the aggregate representation of the entire text prompt. Finally, a learned linear projection (W_{proj}) is applied to obtain the final text embedding:

$$\mathbf{a_c} = \mathbf{W}_{\text{proj}}\, \mathbf{h_c}, \tag{5}$$

which is then normalized to ensure that it lies on the unit hypersphere:

$$\tilde{\mathbf{a}}_\mathbf{c} = \frac{\mathbf{a_c}}{\|\mathbf{a_c}\|}. \tag{6}$$

which serves as the NLP anchor for class c, providing a robust semantic reference that guides the image feature extraction process.

3.5 Warm-Up Training

The warm-up training phase occupies the first 10% of the total training steps and is designed to stabilize the network before more complex loss terms, such as the alignment loss, are introduced. During this phase, the featurizer f and classifier g is updated with cross-entropy loss ℓ_{CE}. At each training step, a minibatch of samples is obtained from the dataset. The inputs $\mathbf{x_i}$ are first concatenated and passed through the feature extractor $f(\cdot)$, implemented using a ResNet-based architecture, to generate latent features. These features are then fed into the classifier $g(\cdot)$ to produce predictions. The classifier is optimized using stochastic gradient descent by minimizing the cross-entropy loss defined as

$$\mathcal{L}_{\text{warm-up}} = \sum_{i=1}^{N} \ell_{CE}\Big(g\big(f(\mathbf{x_i})\big), y_i\Big), \tag{7}$$

where y_i are the true labels and N is the number of samples in the minibatch. In parallel, the moving average network is initialized as a deep copy of the primary network and is updated iteratively to stabilize learning.

Specifically, after a predefined number of iterations, the EMA network parameters $\boldsymbol{\theta}_{\text{ema}}$ are updated as follows:

$$\boldsymbol{\theta}_{\text{ema}} \leftarrow \frac{\boldsymbol{\theta}_{\text{ema}} \cdot \text{ema_count} + \boldsymbol{\theta}}{\text{ema_count} + 1}, \tag{8}$$

where $\boldsymbol{\theta}$ are the parameters of the primary network and ema_count tracks the number of updates. After each update, the EMA network is refreshed according to the update rule described above.

3.6 Weighted Loss

Let $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ denote a batch of N training samples, where $\mathbf{x}_i$ is the ith input feature vector and $y_i \in \{1, 2, \ldots, C\}$ is its corresponding class label for a total of C classes. We denote by $Proj_{y_i}(\cdot)$ the mapping layer corresponding to label y_i that projects feature embeddings into a semantic space aligned with NLP-derived anchors, and by λ a hyperparameter balancing the alignment loss and the classification loss.

Alignment Loss: One way to enforce that the features align with the corresponding NLP anchor is to minimize the negative cosine similarity loss between the mapped feature representation $Proj_{y_i}(f(\mathbf{x}_i))$ and the fixed NLP anchor $\mathbf{a}_{y_i}$. This loss is defined as

$$\mathcal{L}_{\text{anchor}}(\mathbf{x}_i) = -\frac{Proj_{y_i}(f(\mathbf{x}_i)) \cdot \mathbf{a}_{y_i}}{\|Proj_{y_i}(f(\mathbf{x}_i))\| \, \|\mathbf{a}_{y_i}\|}. \tag{9}$$

By minimizing $\mathcal{L}_{\text{anchor}}$, we effectively maximizes the cosine similarity between the projected features and the NLP anchor. This process ensures that the learned features are directionally aligned with the semantic anchors, ultimately promoting representations that are both semantically meaningful and domain-invariant.

Weight Computation: To prioritize high-confidence training instances, we assign importance weights to samples based on their similarity to the NLP anchors. We introduce a temperature parameter τ to adjust the sharpness of the softmax weighting distribution—higher values of τ yield a more peaked distribution that emphasizes strongly aligned samples, whereas lower values produce a softer weighting. Specifically, we first compute the softmax weights over the alignment costs:

$$w_i \;=\; \frac{\exp\Big(-\,\tau\, L_i\Big)}{\sum_{j=1}^{N} \exp\Big(-\,\tau\, L_j\Big)}, \tag{10}$$

where $L_i = -\cos\Big(Proj_{y_i}(f(\mathbf{x}_i)),\, \mathbf{a}_{y_i}\Big)$. Rewriting this, we obtain:

$$w_i \;=\; \frac{\exp\Big(\tau\, \cos\big(Proj_{y_i}(f(\mathbf{x}_i)),\, \mathbf{a}_{y_i}\big)\Big)}{\sum_{j=1}^{N} \exp\Big(\tau\, \cos\big(Proj_{y_j}(f(\mathbf{x}_j)),\, \mathbf{a}_{y_j}\big)\Big)}. \tag{11}$$

Overall Loss: To maintain classification accuracy while leveraging the selective sampling, we incorporate a weighted cross-entropy loss, where samples with higher w_i values contribute more significantly to the overall loss. The final loss function combines the alignment loss and the weighted cross-entropy loss:

$$\mathcal{L} \;=\; \lambda \sum_{i=1}^{N} w_i \Big[-\,\cos\big(Proj_{y_i}(f(\mathbf{x}_i)),\, \mathbf{a}_{y_i}\big)\Big] \;+\; \sum_{i=1}^{N} w_i\, \ell_{CE}\Big(g\big(f(\mathbf{x}_i)\big),\, y_i\Big) \tag{12}$$

The overall loss function serves a dual purpose. The alignment loss minimizes the negative cosine similarity between the projected features and their corresponding NLP anchors, compelling the model to learn representations that are semantically consistent with robust class prototypes. This semantic alignment acts as a natural filter for noise—noisy or out-of-distribution samples tend to have representations that poorly align with their class anchors. Consequently, when the cross-entropy loss minimizes classification error, it focuses primarily on high-confidence samples that are well-aligned. Thus, by jointly minimizing both alignment and

classification errors, the loss function not only ensures accurate predictions but also inherently suppresses the adverse influence of noisy data.

Reformulating the equation above, we obtain

$$\mathcal{L} \;=\; \sum_{i=1}^{N} w_i \left[\lambda \Big(-\cos\big(Proj_{y_i}(f(\mathbf{x}_i)),\, \mathbf{a}_{y_i}\big) \Big) \;+\; \ell_{CE}\Big(g\big(f(\mathbf{x}_i)\big),\, y_i \Big) \right] \tag{13}$$

From this formulation, the weight term w_i acts as an importance factor that regulates the influence of each sample on the overall loss. This regularization mechanism ensures that both the alignment loss and the classification loss emphasize reliable, high-confidence data. By down-weighting the contribution of samples that are likely noisy or out-of-distribution, the weight term effectively minimizes the adverse impact of noise during training. Consequently, this leads to more robust feature learning and improved classification accuracy.

3.7 Iterative Update

After the warm-up, the full A^3W optimization procedure begins. We alternate between updating 1) the featurizer and classifier to improve classification performance; 2) the mapping layers M_{y_i} to enhance alignment with NLP anchors. The classifier and featurizer are updated in every step, but the mapping layers are only updated at specific intervals to prevent overfitting and to maintain stability. The mapping layer updates occur when:

$$\left(\frac{\text{step}}{\text{steps_per_epoch}} \right) \mod \left(\frac{\text{n_steps}}{\text{steps_per_epoch} \times 10} \right) = 0. \tag{14}$$

which ensures that the mapping layers are updated once every 10% of the total training steps. The update loss uses:

$$\mathcal{L} = \lambda \sum_{i=1}^{N} \mathbf{w_i} L_i + \sum_{i=1}^{N} \mathbf{w_i} \ell_{CE}(g(f(\mathbf{x_i})), y_i), \tag{15}$$

where w_i are softmax-scaled importance weights presented above. The training procedure is summarized in Algorithm 1. Each training step consists of sampling a mini-batch from the dataset, computing both the alignment loss and weighted cross-entropy loss, and updating either the mapping layers or the featurizer and classifier based on the step schedule. The algorithm continues until convergence or until the predefined number of steps is reached.

Algorithm 1 Training Outline for A^3W

Require: Dataset $\mathcal{D}$ with classes $\{1, \ldots, C\}$, hyperparameters $\lambda, \tau, \ldots$

Initialize: featurizer $f(\cdot)$, classifier $g(\cdot)$, empty mapping layers $\{Proj_1, \ldots, Proj_C\}$
Set NLP anchors: $\{\mathbf{a_1}, \ldots, \mathbf{a_C}\}$ via CLIP (Algorithm invokes set_nlp_anchor)
Warm-up Training:
for 10% of steps **do**
  Sample mini-batch $\{(\mathbf{x_i}, y_i)\}$ from $\mathcal{D}$
  Update parameters with $\mathcal{L}_{\text{warm-up}}$
end for
Main Training:
for step = 1 to n_steps **do**
  Sample mini-batch $\{(\mathbf{x_i}, y_i)\}$ from $\mathcal{D}$
  if *condition for maplayer update is met* **then**
    Update mapping layers with $\mathcal{L}$
  else
    Update featurizer and classifier and layers with $\mathcal{L}$
  end if
end for

3.8 Theoretical Insights

Our algorithm is designed to mitigate the adverse effects of label noise and domain shifts by integrating several key components, as formalized by our previously defined equations. Our goal is to learn a predictor $h\colon \mathcal{X} \to \mathcal{Y}$ that minimizes the error on $\mathcal{D}_T$. Define the *expected error* of h on domain $\mathcal{D}_i$ as

$$\epsilon_{P_d}(h) \;=\; \mathbb{E}_{(x,y)\sim P_d}\big[\mathbf{1}\{h(x) \neq y\}\big], \tag{16}$$

where P_d denotes the joint probability distribution over the input space $\mathcal{X}$ and label space $\mathcal{Y}$ for domain $\mathcal{D}_i$, capturing both the inherent variability of the data and the process by which labels may be corrupted (i.e., y may be flipped to $\tilde{y}$ with probability p). When p is high, standard empirical risk minimization can easily overfit to spurious correlations in the noisy labels. To counteract this, the alignment loss (see Eq. equation 9) serves as a semantic prior that constrains the feature extractor f to produce representations that lie close to the fixed semantic anchors. This regularizing effect of the alignment loss can be formalized as follows:

Lemma 3.1 (Semantic Prior Restricts Hypothesis Space)**.** *Suppose $\mathcal{H}$ is the space of predictors induced by $(f, \{Proj_c\})$. If $\max_{x,c} \|\nabla \mathcal{L}_{anchor}(x,c)\| \leq \gamma$, then $\mathcal{H}$ excludes functions whose representations deviate from the anchors by more than a constant factor related to γ. In particular, spurious correlations that push the embeddings away from these anchors become suboptimal.*

Proof. By definition,

$$\nabla \mathcal{L}_{\text{anchor}}(x,c) = -\nabla \cos\Big(Proj_c(f(x)), \mathbf{a_c}\Big). \tag{17}$$

A uniformly bounded gradient implies that local changes in $f(x)$ away from $\mathbf{a}_c$ incur non-negligible costs. Hence, any hypothesis that aligns poorly with anchors sees a high penalty, effectively restricting the feasible region of $\mathcal{H}$. □

Lemma 3.1 shows that anchor alignment behaves like a regularizer, steering the network away from memorizing noise-laden features. To further enhance robustness, we employ a continuous weighting scheme (Eq. equation 11) that assigns higher importance to samples with strong semantic alignment. This reweighting mechanism effectively adjusts the empirical risk, as captured in our overall loss function (Eq. equation 12), so that samples likely to be correctly labeled have a greater influence during training.

Theorem 3.1 (Robustness under Weighted ERM)**.** *Let the learned hypothesis $h\colon \mathcal{X} \to \mathcal{Y}$ be defined as*

$$h(x) = g(f(x)), \tag{18}$$

where f is the feature extractor and g is the classifier. Suppose that a fraction α of the training samples in domain $\mathcal{D}_i$ are corrupted. Then, if the temperature parameter τ is sufficiently large, the softmax weights w_i computed via Eq. equation 11 will concentrate on the uncorrupted samples. This concentration effectively reweights the empirical distribution to approximate a clean distribution, such that the overall risk via Eq. equation 12 approximates the risk on noise-free data. Consequently, the learned hypothesis h is less prone to overfitting to label noise and spurious correlations.

Proof. When τ is large, the exponential function in equation 11 amplifies differences in the alignment cost. If a corrupted sample $(x_i, \tilde{y}_i)$ has an incorrect label, the alignment cost $\mathcal{L}_{\text{anchor}}(x_i, \tilde{y}_i)$ tends to be higher (poorer alignment). Hence, w_i becomes small. This effectively filters out mislabeled samples from dominating the training objective, approximating the scenario of training on mostly correct labels. □

Theorem 3.1 indicates that our weighting strategy can mitigate the detrimental effects of noise, enabling the model to approximate the true (clean) distribution more closely. These results not only demonstrate the efficacy of our reweighting strategy in handling noisy labels but also motivate a further examination of how our approach reduces the discrepancy between noisy and clean distributions. Let P and Q be the distributions of the source and target domains, respectively, with label noise in P. Many domain generalization results rely on bounding a distributional divergence $\text{div}(P, Q)$. Noise can inflate this divergence by altering label proportions or feature-label mappings. However, semantic alignment and selective reweighting encourage the model to focus on consistent semantic cues, thereby lowering the *effective* divergence to the clean distribution.

Formally, define the weighted empirical distribution

$$\widehat{P} \;=\; \sum_{i=1}^{N} w_i \delta_{(x_i, y_i)}, \tag{19}$$

where δ denotes the Dirac measure and each sample (x_i, y_i) is reweighted by its importance w_i. In other words, instead of treating each sample equally as in standard empirical risk minimization, we solve:

$$\min_{h \in \mathcal{H}} \; \mathbb{E}_{(x,y) \sim \widehat{P}} \big[\ell\big(h(x), y\big) \big] \;=\; \min_{h \in \mathcal{H}} \sum_{i=1}^{N} w_i \, \ell\big(h(x_i), y_i\big). \tag{20}$$

This reweighting mechanism effectively suppresses the influence of corrupted samples in the empirical risk minimization process. As a result, the reweighted distribution $\widehat{P}$ becomes a closer approximation of the noise-free (clean) distribution. Formally, by minimizing the risk under $\widehat{P}$, the learned hypothesis h is more likely to reflect the patterns present in the clean data rather than the spurious correlations induced by noise. In other words, by down-weighting noisy samples, our approach reduces the divergence between the empirical distribution $\widehat{P}$ and the true clean distribution Q, i.e., $\mathrm{div}(\widehat{P}, Q)$ is reduced. This reduction in discrepancy implies that the model is trained on a distribution that better represents the underlying data, ultimately leading to improved generalization performance on the unseen target domain.

Table 1: Dataset Information for Domain Generalization Benchmarks

Dataset	**Domains**	**# Classes**	**Class Descriptions**	**# Images**
PACS	Photo, Art, Cartoon, Sketch	7	Dog, Elephant, Giraffe, Guitar, Horse, House, Person	9,991
VLCS	Caltech101, LabelMe, SUN09, VOC2007	5	Bird, Car, Chair, Dog, Person	10,729
OfficeHome	Art, Clipart, Product, Real	65	Office/home objects	15,500
SVIRO	4 Car Makes	7	Description of back seat	25,000
DomainNet	Clipart, Infograph, Painting, Quickdraw, Real, Sketch	15	Common Objects	25,730

4 Experiments

4.1 Experimental Setup and Datasets Preprocessing

We conducted our experiments on a server with 10 NVIDIA Quadro RTX 6000 24G GPUs. We extensively evaluate our method on domain generalization with noisy data using four benchmark datasets. For more details, please refer to their original publications. **PACS** (Li et al., 2017) is a 7-class image classification dataset that spans four distinct domains (Photo, Art Painting, Cartoon, and Sketch). Renowned for its diverse artistic styles, PACS offers a challenging testbed for robust representation learning. **VLCS** (Fang et al., 2015) comprises 5 classes drawn from four domains (Caltech101, LabelMe, SUN09, and VOC2007). Each domain originates from a different source, resulting in significant variability in image characteristics and presenting a broad generalization challenge with both natural and scene-centric images. **Office-Home** (Venkateswara et al., 2017) is a 65-class dataset designed to capture common objects in everyday office and home environments. Its wide range of object categories and the substantial variation in style and background make it a rigorous benchmark for domain generalization. **SVIRO** (Cruz et al., 2020) is a synthetic dataset focused on vehicle interiors. It contains 25,000 images from 10 distinct vehicle interior environments and features 7 occupant classes. SVIRO provides a challenging scenario for domain generalization, especially in handling variations in interior design and occupancy. **DomainNet** (Peng et al., 2019) is a large-scale dataset spanning six domains (Clipart, Infograph, Painting, Quickdraw, Real, Sketch) with 345 classes. It encompasses a broad range of styles and object categories, capturing significant distribution shifts across these domains. Due to computation resource limitation, we only used the first four domains in SVIRO and 15 classes per domain for DomainNet. More information is provided in 1.

All experiments are conducted using DomainBed (Gulrajani & Lopez-Paz, 2020) to ensure consistency across datasets. To assess the out-of-distribution performance of various algorithms, we inject instance-independent symmetric label noise based on (Qiao & Low, 2024) where we add 10% and 25% label noise. For all presented

Table 2: Cross-test accuracy (%) for domain shifts under a noise level of $\eta = 0.25$. Best results in **bold**.

Method	PACS	VLCS	Office-Home	SVIRO	DomainNet
ERM	74.5 ± 0.6	71.9 ± 0.6	54.9 ± 0.3	77.9 ± 0.9	65.0 ± 0.4
GroupDRO	74.7 ± 0.4	71.2 ± 0.2	54.1 ± 0.3	75.4 ± 0.5	68.4 ± 0.2
IRM	71.0 ± 1.9	70.3 ± 0.3	53.9 ± 1.8	62.7 ± 14.7	37.6 ± 15.5
VREx	73.5 ± 0.7	71.8 ± 0.6	53.0 ± 0.9	73.6 ± 2.6	64.9 ± 0.8
Mixup	75.2 ± 0.9	71.9 ± 0.5	57.5 ± 0.3	83.3 ± 1.5	67.9 ± 0.2
A^3W **(ours)**	$\mathbf{82.1 \pm 0.5}$	$\mathbf{76.1 \pm 0.3}$	$\mathbf{65.2 \pm 0.2}$	$\mathbf{93.9 \pm 0.1}$	$\mathbf{73.9 \pm 0.2}$

real-world datasets, we employ a ResNet-50 (He et al., 2016) pretrained on ImageNet (Deng et al., 2009), and standard data augmentation techniques are applied across all experiments. For the domain-shift datasets, we used the same metric as (Qiao & Low, 2024), in which we perform single-domain cross-test experiments by designating each domain in turn as the test set and using the remaining domains for training. In these experiments, we use 20% of the test data for model selection, and no early stopping is applied. For each dataset, we run 3 independent trials; in each trial, we perform a hyperparameter search over 20 different configurations according to DomainBed's default settings. Each environment were made to be the target dataset independently, thus 240 total trials were performed for each dataset. Specifically, we train all models for 5000 steps on all real-world datasets.

4.2 Baseline Methods

We compare A^3W against a suite of representative **baseline methods** from the domain generalization literature (Qiao & Low, 2024):

- *ERM*: Empirical Risk Minimization, which trains a single model on the aggregated source domains (Vapnik, 1991).
- *Mixup*: Mixes pairs of source samples to create interpolated training examples, improving robustness (Zhang et al., 2017).
- *GroupDRO*: Optimizes worst-group loss among source domains to handle distribution shifts (Sagawa et al., 2019).
- *IRM*: Invariant Risk Minimization, which enforces domain-invariant representations to improve out-of-distribution generalization (Arjovsky et al., 2019).
- *V-REx*: Variance Risk Extrapolation, which penalizes variance in risk across different environments to improve generalization (Krueger et al., 2021).

We obtain the comparison results from Qiao & Low (2024), and we use the Pytorch suite developed by Gulrajani & Lopez-Paz (2020) for base for algorithm implementation. The subset of TerraIncognita from Qiao & Low (2024) was no longer publicly available, so we used the SVIRO dataset. The SVIRO dataset's baseline method result was obtained by running the experiment in the same setup as Qiao & Low (2024). The target domain is divided into validation and test sets. The validation set is used to tune the hyperparameters by selecting the configuration that yields the highest accuracy. Using the optimal hyperparameter configuration, we then evaluate the model's performance on the test set, and the average classification accuracy on the target domain is used as the evaluation metric.

Table 3: Cross-test accuracy (%) for domain shifts under a noise level of $\eta = 0.1$. Best results in **bold**.

Method	PACS	VLCS	Office-Home	SVIRO	DomainNet
ERM	82.0 ± 0.5	75.0 ± 0.3	62.2 ± 0.1	80.8 ± 1.9	70.8 ± 0.5
GroupDRO	82.4 ± 0.3	75.1 ± 0.1	61.3 ± 0.4	78.6 ± 2.1	71.9 ± 0.1
IRM	80.4 ± 1.2	74.6 ± 0.3	61.2 ± 1.2	72.5 ± 8.1	38.9 ± 19.8
VREx	81.4 ± 0.2	75.0 ± 0.1	60.6 ± 0.5	81.6 ± 3.5	66.9 ± 0.4
Mixup	83.6 ± 0.1	75.5 ± 0.2	63.9 ± 0.1	84.1 ± 0.7	69.2 ± 0.6
A^3W **(ours)**	$\mathbf{85.2 \pm 0.2}$	$\mathbf{78.5 \pm 0.2}$	$\mathbf{68.2 \pm 0.3}$	$\mathbf{94.8 \pm 0.3}$	$\mathbf{76.1 \pm 0.1}$

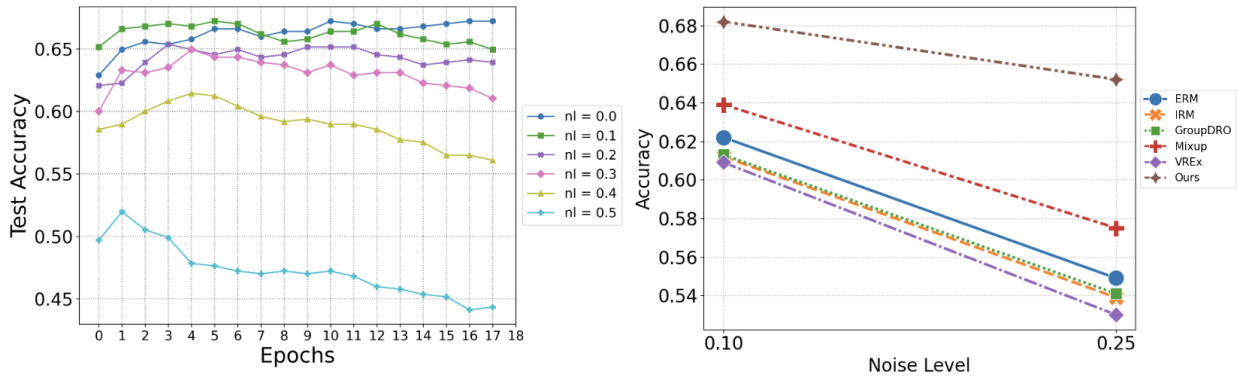

Figure 3: Noise analysis. (a) The effect of increasing noise levels on classification accuracy reveals that higher noise leads to a sharper decline, reflecting an increased tendency to overfit. (b) A^3W is most robust to noise injection, with its accuracy decreasing by only 0.2 when noise increases from 0.1 to 0.25, in contrast to other algorithms, which show declines between 0.427 and 0.527.

4.3 Experimental Results

Table 2 and 3 present the average test accuracy for different values of $\eta = 0.1$ and $\eta = 0.25$, respectively. The experimental results show that A^3W consistently outperforms all baseline methods. We observe the following:

1) A^3W consistently achieves the highest accuracy across all domains in all datasets. Its margin of improvement often exceeds 6–8% on average compared to the baseline methods, indicating stable training dynamics, leading to consistent accuracy improvements across multiple datasets.

2) A^3W shows the strongest performance gain in SVIRO (averaged to 13%), where the most semantic information is given, proving the importance of the NLP anchor in guiding the sampling process of learning the featurizer.

3) Among the baselines, **Mixup** frequently attains the second-best performance, suggesting that interpolation-based augmentation can help mitigate moderate noise. However, its improvements are relatively small and are inconsistent across different domains.

4) In our noise analysis (see Figure 3), we observe that the general training accuracy tends to drop earlier as noise levels increase, a sign of overfitting to noise. Among all evaluated methods, A^3W exhibits the smallest decline in accuracy with rising noise, demonstrating its resilience to overfitting under noisy conditions. Additionally, the flexibility of simple loss reweighting in handling noisy labels allows it to adapt well to various network architectures. Thus, in subsequent experiments, we use A^3W as the default approach unless otherwise specified.

Table 4: Cross-test accuracy (%) for domain shifts. Best results in **bold**.

Method	PACS	VLCS	Office-Home	SVIRO	DomainNet
w/out NLP anchor	80.7	75.5	65.1	91.7	85.5
w/out weighted loss	79.7	74.2	64.7	93.6	85.8
A^3W **(baseline)**	**82.1**	**76.1**	**65.2**	**93.9**	**86.5**

4.4 Ablation Study

To better understand the role of each component in A^3W, we conduct ablation studies under three key questions, as summarized in Table 4. The experiments span multiple datasets, and the results highlight how removing NLP anchor alignment or omitting the softmax weighting mechanism impacts overall performance.

1) Why does removing NLP anchor alignment reduce performance? In Table 4, discarding the NLP anchor alignment step (i.e., "w/out NLP anchor") yields an average accuracy drop of about 0.86% across the listed datasets. This decrease arises because anchor alignment provides semantic guidance that helps the model differentiate meaningful features from spurious correlations. Without alignment, the feature extractor lacks an external semantic reference, making it more prone to overfitting on noisy labels and domain-specific artifacts.

2) How does eliminating softmax weights affect training stability? When we replace the adaptive softmax weighting with uniform weights (i.e., "w/out weighted loss"), the accuracy declines by an average of 1.16% across the datasets. This finding suggests that the continuous weighting scheme (Section 3.8) is crucial for emphasizing well-aligned (likely clean) samples. In contrast, uniform weighting fails to down-weight noisy or misaligned samples, reducing training stability and overall performance.

3) Are both alignment and weighting necessary for robust generalization? Finally, the baseline A^3W model—which integrates both NLP anchor alignment and softmax weighting—achieves the highest accuracy on average (e.g., 80.76% on PACS). These results indicate that semantic anchors and adaptive weighting reinforce each other: alignment ensures meaningful feature extraction, while the weighting mechanism selectively highlights reliable data. Removing either component leads to noticeable performance degradation, underscoring their combined importance for robust domain generalization under noisy labels.

These findings confirm that semantic alignment with textual anchors (and the associated softmax weighting) plays a pivotal role in boosting generalization. The additional MA network contributes incremental stability but is not solely responsible for the performance gains.

4.5 Convergence Analysis and Feature Clustering

In addition to the experimental results, we further assess the feasibility of our approach by examining its trainability and convergence properties. Figure 4 illustrates the convergence behavior of our model as the training step size increases. The convergence trajectories, shown under three different random seeds and training configurations across two datasets, demonstrate that A^3W achieves stable optimization and effectively minimizes the overall loss—even in the presence of label noise. This stability validates our method's robustness and its ability to learn domain-invariant representations. Furthermore, Figure 5 presents t-SNE embeddings of features learned by three different methods on the PACS dataset. In these visualizations, each point represents a sample, with color indicating the class and marker shape indicating the domain. By comparing the best models from each approach, it is evident that the proposed A^3W produces more cohesive and well-separated clusters across both training and test domains, underscoring its superior capability in learning discriminative features.

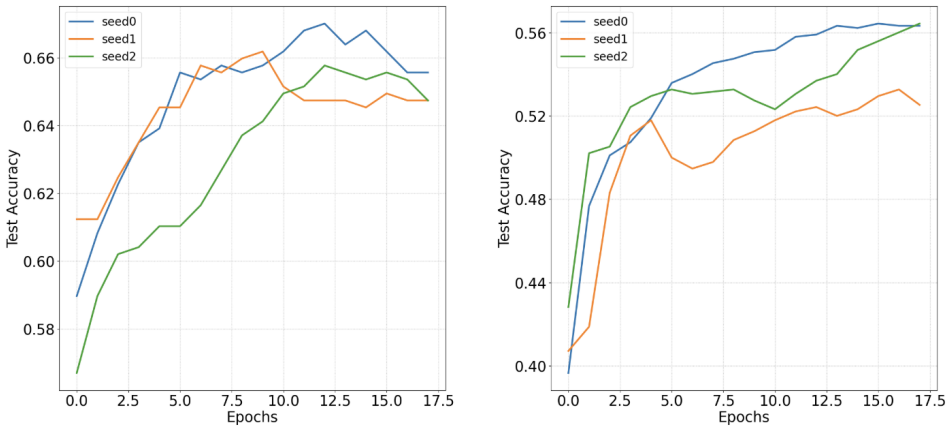

Figure 4: Convergence trajectories under three different random seeds and training configurations for two datasets.

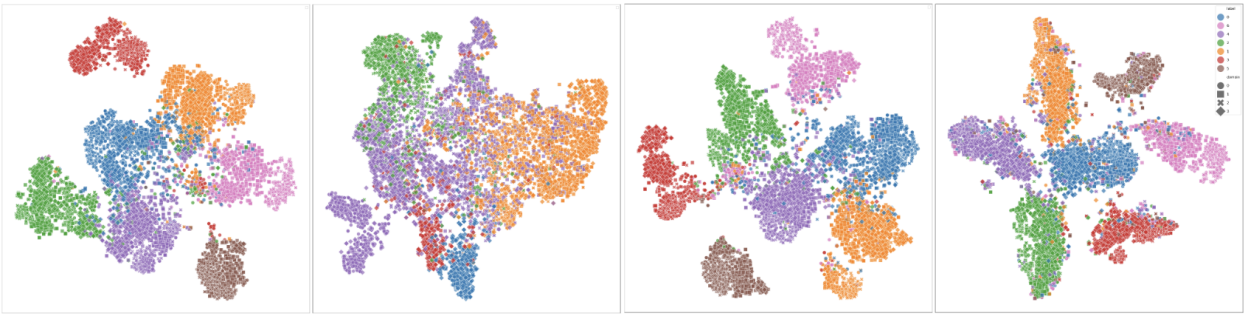

Figure 5: t-SNE embeddings of three methods on the PACS dataset. From left to right are ERM, IRM, MixUp, and A^3W.

4.6 Parameter Sensitivity

We consider four key hyperparameters in our method: the regularization parameter (λ), the iteration frequency, the temperature (τ), and the learning rate (lr). For fairness, we keep all other settings fixed while varying one parameter at a time, and we set the random seed for each trial. Figure 6 presents the parameter sensitivity analysis, with red markers indicating the optimal performance points. The best performance is achieved at $\lambda = 0.1$, though the accuracy remains competitive even when λ deviates slightly from this value, demonstrating resilience to parameter shifts. Similarly, increasing the iteration frequency from 5 to 20 results in only minor changes in accuracy, which underscores the method's stability. For the temperature, varying τ from 5 to 15 shows that the optimal performance is attained at $\tau = 10$, with a sharp decline in accuracy observed when τ is raised to 15. This suggests that an excessive temperature causes the amplification in the softmax to become overly aggressive, rendering it too selective and potentially discarding useful information; however, when properly tuned, it also helps to effectively diminish the effect of samples that deviate from the semantic anchors. Although Theorem 3.1 guarantees that a sufficiently high temperature will favor uncorrupted data through the softmax weighting, the key is finding the right balance–high enough to filter out noise without overshooting. The proposed method ("Ours") achieves the highest accuracy at 10^{-4}, as indicated by the red cross mark, but performance drops significantly at higher learning rates. Both ERM and Mixup exhibit more stable performance across lower learning rates but also experience a sharp decline at 10^{-3} and beyond. Overall, these findings demonstrate that our approach exhibits robustness across a broad spectrum of hyperparameter settings, maintaining strong performance even under suboptimal conditions. Furthermore, they suggest that careful hyperparameter tuning can further improve performance, offering a competitive advantage over alternative methods.

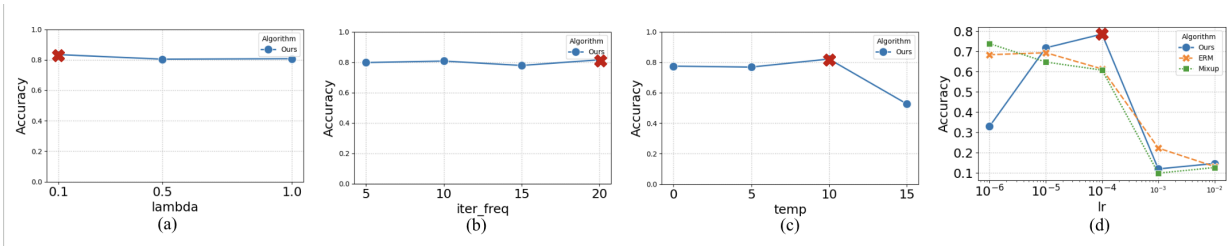

Figure 6: Parameter sensitivity analysis.

4.7 Impact of Semantic Richness in NLP Anchors

Our ablation studies reveal that NLP anchors provide effective guidance, a finding further supported by experimental results on the SVIRO dataset (see Table 2 and Table 3). In particular, when descriptive class names such as "car back seat," "infant car seat," and "child in convertible car seat" were used, A^3W achieved the highest performance improvement among all datasets. In contrast, other datasets used single-word labels showed less pronounced gains. However, this reliance on rich semantic information may pose challenges when the categories are very similar or less frequently encountered. We ruled out the possibility that the number of classes was a limiting factor—our experiments on the OfficeHome dataset, which includes 65 classes, still demonstrated performance gains over the baselines. Future work could explore more tailored network architectures for instance-specific domain generalization, especially in scenarios involving closely related classes.

4.8 Training Noise as an Implicit Regularizer

In some parameter settings, we observed that introducing more noise during training can actually lead to improved accuracy, as shown in Fig. 3. This counterintuitive result occurs because additional noise acts as a form of regularization. Essentially, it prevents the model from overfitting to spurious correlations and irrelevant details in the training data. Previous work (Chen et al., 2023) suggests that this injected noise encourages the network to learn more robust and invariant features by smoothing the loss landscape and promoting the discovery of flatter minima. As a result, even though the training data is noisier, the model is better able to capture the underlying patterns that generalize well to the target domain. This leads to improved performance on unseen data, as the network becomes less sensitive to the peculiarities of the noisy training samples.

4.9 Cosine Similarity Loss for Feature Representation Learning

Negative cosine similarity is commonly used as a loss function when learning feature representations, especially in tasks such as metric learning, similarity-based learning, and contrastive learning. We compared L2 and cosine similarity, where cosine similarity greatly outperformed L2 loss. Although both encourage the learning of representations that are close together for similar items (i.e. samples from the same class) and far apart for dissimilar ones to support the formation of a more discriminative and robust feature space, cosine similarity captures the orientation of the vectors rather than their magnitude. In other words, cosine similarity is scale invariant: if a vector is scaled by a constant factor, its cosine similarity with another vector remains unchanged, whereas L2 loss is directly affected by changes in magnitude. This property is particularly advantageous in tasks where the relative direction of feature vectors (which captures semantic information) is more important than their absolute values. Additionally, cosine similarity naturally emphasizes the alignment between vectors, encouraging features from the same class to be directionally similar, regardless of their scale. This can lead to more robust clustering and improved generalization, especially when the features may vary in magnitude due to factors unrelated to the underlying semantics.

5 Discussion

Despite these promising results, A^3W has several limitations that can be improved. The computational cost of A^3W is another aspect worth considering. Compared to baseline DG methods such as ERM or CORAL, our approach introduces additional *mapping layers* for each class and computes cosine similarity loss with softmax-based weighting, increasing both parameter count and training time. While our experiments indicate that the additional overhead is manageable (approximately a 10% increase in runtime), scaling A^3W to large-scale datasets with hundreds of classes could become computationally expensive—particularly when replacing simple modules with larger ones (e.g., cross attention), as evidenced by the significant time cost observed over 240 trials. Potential solutions include optimizing the mapping layers via *low-rank factorization* or *class clustering* to reduce redundancy in the alignment process. Another promising direction for improvement involves *adaptive anchor selection*. Currently, A^3W assumes a fixed set of text embeddings throughout training, which may not always be optimal. Instead of static anchors, one could explore *dynamic anchor refinement*, where text embeddings evolve based on learned feature distributions. This strategy would enable the model to refine its representations over time, thereby improving alignment with the most semantically relevant concepts. Additionally, integrating *contrastive learning* techniques could further strengthen the alignment between visual and textual representations. Future work could also extend A^3W to *multi-label and hierarchical classification* settings. Many real-world applications involve objects that belong to multiple categories simultaneously (e.g., an image of a wolf could be categorized as both "canine" and "wild animal"). Current A^3W anchors operate at the class level, but incorporating hierarchical category embeddings (e.g., WordNet-based representations) could improve generalization by capturing higher-level semantic relationships. Similarly, expanding the algorithm to support multi-modal domain generalization—incorporating additional cues such as audio or structured metadata—could further enhance robustness. Finally, another promising extension involves *domain-aware prompts*. Instead of using generic text descriptions (e.g., "*a photo of a {class}*"), dynamically tailoring prompts to reflect domain characteristics (e.g., "*a sketch of a dog*" in PACS) could improve alignment. This would enable a more fine-grained adaptation of NLP anchors to specific domain shifts, further mitigating performance degradation in unseen domains.

6 Conclusion

In this paper, we introduced A^3W, a simple yet effective domain generalization algorithm that integrates external knowledge from large-scale language models to guide visual feature learning. By mapping learned representations to class-specific NLP anchors, we impose additional semantic constraints that mitigate domain-specific biases. Experimental results across five benchmark datasets demonstrate consistent improvements over state-of-the-art methods, underscoring the effectiveness of *knowledge-guided* strategies in dealing with unseen domains to the best of our knowledge.

Moreover, our ablation studies reveal that both semantic alignment and adaptive weighting play pivotal roles in enhancing the robustness of our approach. Notably, our algorithm's modular design allows for easy replacement or customization of its individual components—such as integrating different moving average strategies or alternative weighting schemes—to better suit diverse tasks. This flexibility not only simplifies the process of adapting to new challenges but also sets the stage for future explorations in knowledge-guided domain generalization. As vision-language models continue to evolve, the potential to further refine and extend these methods promises even greater strides toward interpretable, robust, and versatile deep learning systems.

References

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 456–473, 2018.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79:151–175, 2010.

Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2229–2238, 2019.

Hao Chen, Jindong Wang, Ankit Shah, Ran Tao, Hongxin Wei, Xing Xie, Masashi Sugiyama, and Bhiksha Raj. Understanding and mitigating the label noise in pre-training on downstream tasks. *arXiv preprint arXiv:2309.17002*, 2023.

Steve Dias Da Cruz, Oliver Wasenmuller, Hans-Peter Beise, Thomas Stifter, and Didier Stricker. Sviro: Synthetic vehicle interior rear seat occupancy dataset and benchmark. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 973–982, 2020.

Tirtharaj Dash, Sharad Chitlangia, Aditya Ahuja, and Ashwin Srinivasan. A review of some techniques for inclusion of domain-knowledge into deep neural networks. *Scientific Reports*, 12(1):1040, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. *Advances in neural information processing systems*, 32, 2019.

Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. From captions to visual concepts and back. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1473–1482, 2015.

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

Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pp. 2551–2559, 2015.

Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *International conference on learning representations*, 2017.

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

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International conference on machine learning*, pp. 2304–2313. PMLR, 2018.

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *International conference on machine learning*, pp. 5815–5826. PMLR, 2021.

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

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020.

Geng Liu and Yuxi Wang. Tdg: Text-guided domain generalization. *arXiv preprint arXiv:2308.09931*, 2023.

Fangrui Lv, Jian Liang, Shuang Li, Bin Zang, Chi Harold Liu, Ziteng Wang, and Di Liu. Causality inspired representation learning for domain generalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8046–8056, 2022.

Shuran Ma, Weiying Xie, Daixun Li, Haowei Li, and Yunsong Li. Reducing spurious correlation for federated domain generalization. *arXiv preprint arXiv:2407.19174*, 2024.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International conference on machine learning*, pp. 10–18. PMLR, 2013.

Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8690–8699, 2021.

Nitika Nigam, Tanima Dutta, and Hari Prabhat Gupta. Impact of noisy labels in learning techniques: a survey. In *Advances in Data and Information Sciences: Proceedings of ICDIS 2019*, pp. 403–411. Springer, 2020.

Kento Nishi, Yi Ding, Alex Rich, and Tobias Hollerer. Augmentation strategies for learning with noisy labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8022–8031, 2021.

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

Rui Qiao and Bryan Kian Hsiang Low. Understanding domain generalization: A noise robustness perspective. In *Proceedings of the International Conference on Reinforcement Learning (ICRL 2024)*, 2024.

Bin Qin, Jiangmeng Li, Yi Li, Xuesong Wu, Yupeng Wang, Wenwen Qiang, and Jianwen Cao. Revisiting spurious correlation in domain generalization. *arXiv preprint arXiv:2406.11517*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Md Mahbubur Rahman, Ira Ceka, Chengzhi Mao, Saikat Chakraborty, Baishakhi Ray, and Wei Le. Towards causal deep learning for vulnerability detection. In *Proceedings of the IEEE/ACM 46th international conference on software engineering*, pp. 1–11, 2024.

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Olawale Salaudeen and Oluwasanmi Koyejo. Target conditioned representation independence (tcri); from domain-invariant to domain-general representations. *arXiv preprint arXiv:2212.11342*, 2022.

Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. *arXiv preprint arXiv:1804.10745*, 2018.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE transactions on neural networks and learning systems*, 34(11): 8135–8153, 2022a.

Weiran Song, Zongyu Hou, Weilun Gu, Muhammad Sher Afgan, Jiacheng Cui, Hui Wang, Yun Wang, and Zhe Wang. Incorporating domain knowledge into machine learning for laser-induced breakdown spectroscopy quantification. *Spectrochimica Acta Part B: Atomic Spectroscopy*, 195:106490, 2022b.

Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5552–5560, 2018.

Vladimir Vapnik. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4, 1991.

Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. Learning from noisy large-scale datasets with minimal supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 839–847, 2017.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5018–5027, 2017.

Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *Advances in neural information processing systems*, 31, 2018.

Laura Von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, et al. Informed machine learning–a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):614–633, 2021.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and S Yu Philip. Generalizing to unseen domains: A survey on domain generalization. *IEEE transactions on knowledge and data engineering*, 35(8):8052–8072, 2022.

Na Wang, Lei Qi, Jintao Guo, Yinghuan Shi, and Yang Gao. Learning generalizable models via disentangling spurious and enhancing potential correlations. *IEEE Transactions on Image Processing*, 33:1627–1642, 2024.

Shirley Wu, Mert Yuksekgonul, Linjun Zhang, and James Zou. Discover and cure: Concept-aware mitigation of spurious correlation. In *International Conference on Machine Learning*, pp. 37765–37786. PMLR, 2023.

Shen Yan, Huan Song, Nanxiang Li, Lincan Zou, and Liu Ren. Improve unsupervised domain adaptation with mixup training. *arXiv preprint arXiv:2001.00677*, 2020.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain adaptive ensemble learning. *IEEE Transactions on Image Processing*, 30:8008–8018, 2021.