CRISP: COMPOSITIONAL RELATIONS AS INVARIANT STRUCTURAL PRIORS FOR DOMAIN GENERALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Domain generalization requires identifying stable representations that support reliable classification across domains. Domains may differ in low-level attributes, such as color, texture, or visual style, while preserving the same structural relationships among their underlying components. Existing methods primarily address these differences by improving the training process or aligning features across domains. However, since they leave this shared compositional structure implicit, they may overlook a more reliable source of invariance and consequently generalize less effectively to unseen domains.

We propose **C**ompositional **R**elational **I**nvariance from **S**patial **P**rimitives (CRISP), an image classification framework that factors visual recognition into visual primitives and their relational composition. We represent these compositions using soft unary, binary, and ternary predicates over primitive locations and appearance, yielding differentiable measures of spatial and visual alignment that can be learned end-to-end. To learn primitives and relational structure jointly, we design an end-to-end architecture with three components: (1) a visual backbone that extracts generalized features, (2) a concept bottleneck layer that maps these features to primitive heatmaps with differentiable spatial coordinates, and (3) a structural scoring layer that evaluates candidate spatial relations among the detected primitives. Finally, we compute class probability from the joint evidence of its class-specific relational compositions and localized primitive appearance.

We evaluate CRISP on five real-world image-classification datasets from the widely used DomainBed suite, covering shifts in depiction style, dataset provenance, and camera-trap location. To assess its ability to exploit compositional structure, we additionally evaluate it on CUB-DG, a fine-grained bird-recognition benchmark in which distinguishing species depends on combinations of part-level attributes. Across CUB-DG and the DomainBed benchmark suite, CRISP improves accuracy by a large margin of 7.9 percentage points on CUB-DG and 1.2 percentage points over existing DG methods on DomainBed, achieving the new state-of-the-art on both benchmarks.

1 INTRODUCTION

A central goal in image classification is to learn classifiers that remain accurate beyond the domains on which they are trained. In practice, however, even small changes in camera, lighting, viewpoint, or style can cause a classifier's performance to degrade (Venkateswara et al., 2017; Saenko et al., 2010; Nguyen and Nguyen, 2025; Zhang et al., 2022). Such failures suggest that classifiers often exploit cues that are predictive in the training domain but unstable under distribution shift.

Domain Generalization (DG) has emerged as one of the most promising approaches to this problem, in which a model is trained on multiple labeled source datasets and evaluated on an unlabeled target dataset. Since the target domain is unavailable during learning, a DG method must determine which aspects of the image-to-label mapping are stable enough to transfer.

Relying on strong pretrained backbones (He et al., 2016; Krizhevsky et al., 2012), existing DG methods answer this question by modifying the training objective or optimization procedure. Specifically, this includes aligning feature distributions across domains (Sun and Saenko, 2016; Li et al., 2018), perturbing or mixing samples and feature statistics (Zhang et al., 2018; Zhou et al., 2021;

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Nam et al., 2021), selecting or regularizing solutions that generalize better across domains (Cha et al., 2021; 2022), and exploiting pretrained semantic representations (Shu et al., 2021). Indeed, strong empirical studies show that even carefully tuned ERM remains competitive in standard DG benchmarks (Teterwak et al., 2024).

Although these strategies can be effective, they largely preserve the standard recognition pipeline and rely on backbone representations to capture layout, part structure, and spatial relations implicitly. This leaves structural composition under-specified, despite longstanding evidence of its role in recognition (Biederman, 1987; Felzenszwalb et al., 2009), and may limit performance on compositional domain generalization settings.

As a motivating example, consider images of the same bird species from two domains, Photo and Cartoon. Across these domains, local parts such as the eyes, beak, and wings may change in appearance. For example, they may be rendered with flattened colors, simplified textures, or missing specular highlights. Nevertheless, their coarse spatial organization often remains recognizable: the beak typically lies near the eyes and along the head, while the wings appear in characteristic positions relative to the body. This suggests that a classifier can benefit from distributing evidence between local primitive appearance and compositional structure, rather than forcing primitive detectors alone to absorb domain-specific variation.

In this work, we introduce **C**ompositional **R**elational **I**nvariance from **S**patial **P**rimitives (CRISP), which encourages structural invariance by representing visual categories through spatial relations among learned, localized visual primitives. To express these relations, CRISP uses soft, differentiable spatial predicates: binary predicates capture pairwise constraints, while ternary predicates approximate layout cues such as triangular arrangements and angular constraints. Given these relational predicates, we then represent each class as a sparse set of class-specific spatial compositions, and its score is computed by marginalizing over possible assignments of detected primitives to these compositions. Although related predicates have been used in neuro-symbolic visual reasoning as grounded scene representations (Johnson et al., 2017; Liu et al., 2019; Hsu et al., 2023), our goal is different: we use these predicates not as semantic reasoning targets, but as a differentiable structural inductive bias for domain-generalized classification.

This formulation creates an architectural challenge: CRISP must learn primitive detectors and classification-relevant spatial predicates jointly from image-level supervision. We address this challenge with an end-to-end architecture that learns primitive detectors, structural predicate parameters, and class-specific compositions in a single model. The architecture comprises three components: a visual backbone that extracts visual features, a concept bottleneck layer (Koh et al., 2020) that maps these features to primitive heatmaps with differentiable spatial coordinates, and a structural scoring layer that evaluates candidate spatial relations among the detected primitives. The structural scoring layer encodes the space of possible relational compositions in a differentiable manner, enabling joint optimization of primitive detectors and structural weights via standard gradient descent. The resulting relation scores are aggregated by a final classifier, producing predictions grounded in interpretable compositions of visual primitives and their spatial arrangements.

We evaluate CRISP on the widely used DomainBed benchmark (Gulrajani and Lopez-Paz, 2020) and CUB-DG (Min et al., 2022), a benchmark for compositional domain generalization. On DomainBed, CRISP achieves accuracies of 70.1% and 79.0% with ResNet50 and ViT-B backbones, respectively, surpassing the state-of-the-art method ERM++ (Teterwak et al., 2024) by 1.2 and 0.4 percentage points. On CUB-DG, CRISP yields a larger improvement of 7.9 percentage points with ResNet50, highlighting its effectiveness in targeting compositional domain shift.

We summarize our contributions as follows:

- We propose a structure-aware framework for domain generalization that treats visual categories as compositions of learned primitives and spatial relations, improving performance on compositional recognition under domain shift.
- We introduce an end-to-end differentiable architecture that realizes this framework by learning primitive heatmaps, extracting differentiable spatial coordinates, and scoring class-specific compositions over primitive locations.
- We evaluate CRISP on the DomainBed and CUB-DG benchmark suite, showing that it improves over the prior state of the art on both benchmark suites by 1.2% and 7.9%,

respectively, using the traditional ResNet50 backbone while remaining competitive on the modern ViT-B backbone; our ablation studies further show that the proposed structural scoring layer consistently improves performance across domains and datasets.

The remainder of the paper is organized as follows. Section 2 reviews related work on domain generalization, spatial predicates, and concept learning. Section 3 presents CRISP. Section 4 reports the experimental setup and results. Section 5 concludes with limitations and future directions.

2 Related Work

This section reviews three lines of work most relevant to CRISP: domain generalization, spatial relations and predicates for visual recognition, and concept bottleneck learning.

Domain generalization and adaptation. Domain adaptation and domain generalization both address distribution shift, but differ in target-domain access. Unsupervised domain adaptation assumes unlabeled target samples are available during training, and commonly reduces source-target discrepancy through distribution matching or adversarial objectives (Long et al., 2015; Sun and Saenko, 2016; Ganin et al., 2016; Long et al., 2018). Recent extensions further improve adaptation with transformer architectures, parameter-efficient modules, or vision-language prompting (Sun et al., 2022; Xu et al., 2022; Nguyen and Nguyen, 2025; Bai et al., 2024; Khattak et al., 2023).

Domain generalization is stricter: target domains are unseen during training. Existing DG methods seek transferable representations or training procedures through invariant feature learning, risk or optimization-based regularization, and style or feature perturbation (Muandet et al., 2013; Arjovsky et al., 2019; Krueger et al., 2021; Cha et al., 2021; 2022; Zhou et al., 2021; Nam et al., 2021). While carefully tuned ERM remains a strong baseline on standard DG benchmarks (Gulrajani and Lopez-Paz, 2020), CUB-DG highlights compositional shifts in both appearance and visual structure (Min et al., 2022). CRISP complements these feature-centric approaches by explicitly scoring compositions of learned primitives and differentiable spatial predicates.

Spatial relations and predicates for visual recognition. Spatial structure has long been central to recognition, from classical part-relation theories (Biederman, 1987) and deformable part models (Felzenszwalb et al., 2009) to neural relation modules over objects or regions (Hu et al., 2018). Predicates provide a compact way to express such structure and are widely used in neuro-symbolic visual reasoning for relations such as position, proximity, and containment (Johnson et al., 2017; Mao et al., 2019). Recently, NeSyCoCo studies differentiable composition of predicate scores for compositional generalization in vision-language reasoning (Kamali et al., 2025). In contrast to these works, CRISP uses differentiable spatial predicates not as grounded semantic reasoning targets or language-conditioned programs, but as an inductive bias for domain-generalized classification, providing a vocabulary for scoring spatial compositions among learned visual primitives.

Concept bottleneck learning. Concept bottleneck models predict through intermediate concept representations, often using predefined human-interpretable concepts with concept-level supervision (Koh et al., 2020). Another thread studies concept discovery and concept-based explanations under weaker supervision (Chen et al., 2020). CRISP adopts the bottleneck idea for spatial primitive learning: it learns localized primitive heatmaps from image-level supervision and derives differentiable spatial descriptors from them. Unlike standard concept bottlenecks that classify from attribute-like concept vectors, CRISP uses these descriptors as localized evidence for differentiable relational compositions in the structural scoring layer.

3 Methodology

In this section, we describe the methodology of CRISP. We consider the standard domain generalization setting, in which we are given a training set $\mathcal{D} = \{(X_i, Y_i, d_i)\}_{i=1}^{N}$, where $X_i \in \mathbb{R}^{3\times H\times W}$ is an image, $Y_i \in \mathcal{C}$ is its class label, and $d_i \in \mathcal{S}$ indicates its source domain. At test time, the model is evaluated on samples from an unseen target domain $t \notin \mathcal{S}$ that shares the same label space $\mathcal{C}$ as the source domains but was not observed during training.

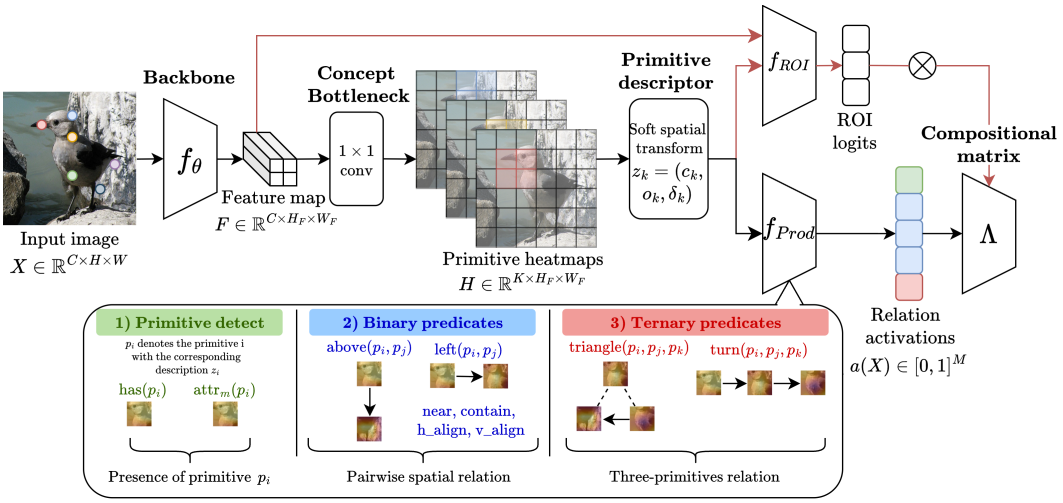

Figure 1: The overall pipeline of CRISP.

As in prior DG work, our goal is to train a classifier that generalizes to unseen target domains. Furthermore, instead of relying solely on backbone representations, we explicitly model visual primitives and their spatial relations, while learning both end-to-end from image-level supervision. We first describe the primitive-and-structure component of CRISP in Section 3.1, where we formally define visual primitives and the spatial predicates over their locations. Then, we describe our network architecture that allows jointly learning both these structures and primitives in Section 3.2. In the same section, we also describe how primitive heatmaps are extracted from backbone features, how spatial predicates are computed over primitive locations in this network, and how we combine both spatial predicates and the backbone features for classification.

3.1 Representing Primitives and Structure

Visual primitives. To represent visual primitives, we assume that there exists a set of K learned visual primitives $\mathcal{P} = \{p_1, \ldots, p_K\}$. Each primitive $p_k : X \mapsto z_k(X)$ is a mapping from any input image X to the corresponding image-dependent descriptor $z_k(X) = \langle c_k(X), \sigma_k(X), \delta_k(X), \ell_k(X) \rangle$, with associated spatial location $c_k(X) \in \mathbb{R}^2$, a presence score $\sigma_k(X) \in [0, 1]$, the corresponding spatial extent $\delta_k(X) \in \mathbb{R}^2$, and finally, a vector for appearance descriptor $\ell_k(X)$. The primitives should be learned from image-level supervision and should not require manual annotations, which we leave for Section 3.2 to describe how p_k and $z_k(X)$ are parameterized and optimized in the full network. Below, we define spatial predicates over these image-dependent primitive descriptors, which are used to form class-specific structural compositions.

Differentiable spatial predicates. Given the set of primitives $\mathcal{P}$, we model their spatial relations as a collection of soft spatial predicates $\mathcal{R}$ with varying arity. Each predicate would takes as input a tuple of primitives $\langle p_1, \ldots, p_r \rangle$ (where r is the arity of the predicate) along with an input image X and output a soft satisfaction score in the range of $[0, 1]$:

$$R : \langle p_1, \ldots, p_r \rangle \times X \mapsto \text{score}_R \in [0, 1]$$

Implementation-wise, all predicates operate over the image-dependent description $z_k(X)$ for each primitive p_k. Thus, for simplicity, in the rest of this section, we omit the dependence on X: We use $z_k = \langle c_k, \sigma_k, \delta_k, \ell_k \rangle$ in place of $z_k(X)$. Furthermore, we expand $c_k = \langle c_k^x, c_k^y \rangle$ and $\delta_k = \langle \delta_k^x, \delta_k^y \rangle$, respectively. For predicates that require box corners, we derive a soft bounding box b_k from the primitive location c_k and extent δ_k:

$$b_k = \langle b_k^{x_1}, b_k^{y_1}, b_k^{x_2}, b_k^{y_2} \rangle = \langle c_k^x - \delta_k^x,\ c_k^y - \delta_k^y,\ c_k^x + \delta_k^x,\ c_k^y + \delta_k^y \rangle.$$

Finally, before defining the predicates, we make two remarks about the parameterization of these predicates. Firstly, all predicate parameters we describe below, including $\kappa_{(\ldots)}$, $m_{(\ldots)}$, $\tau_{(\ldots)}$, $\rho, \beta, \eta, \gamma, \theta$ and ϕ are all learnable. Secondly, we parameterize each predicate by its family; for example, the

above predicate R_{above}'s parameters will be shared across different combinations of input primitives, and thus, the set of learnable parameters for all these predicates is fixed with respect to the number of primitives and the number of classes. We now describe each spatial predicate in detail below.

Primitive presence. Spatial relations describe how primitives are arranged, while appearance cues help distinguish primitives with similar spatial configurations. Since patterns such as curved or dashed contours can recur across different primitives, we introduce a shared vocabulary of N_{attr} learned unary appearance predicates. This vocabulary allows the model to reuse appearance tests across primitives and combine them with different spatial relations. Implementation-wise, for each primitive p_k and its corresponding appearance descriptor $\ell_k \in \mathbb{R}^d$, we apply N_{attr} scalar affine heads to this descriptor:

$$R^{(n)}_{\text{attr}}(p_k) = \text{sigmoid}\big(w_n^\top \text{LayerNorm}(\ell_k) + b_n\big), \qquad n \in \{1, \ldots, N_{\text{attr}}\},$$

where $w_n \in \mathbb{R}^d$ and $b_n \in \mathbb{R}$ are learned parameters. Each vocabulary element m thus applies the same learned appearance test to every primitive, producing a soft predicate activation $R^{(n)}_{\text{attr}}(p_k) \in [0, 1]$. A primitive can activate multiple predicates, allowing its appearance to be represented through combinations of these shared vocabulary elements. The structural layer, as we would describe later, combines these activations with spatial predicates to form class-specific structural compositions.

Binary predicates. We use binary predicates to encode pairwise spatial relations between two primitives p_i and p_j. They correspond to common spatial relations used in predicate-based visual reasoning, such as relative position, alignment, proximity, and containment Johnson et al. (2017); Yi et al. (2018); Mao et al. (2019). We give their detailed formula below:

$$R_{\text{above}}(p_i, p_j) = \text{sigmoid}\big(\kappa_\uparrow(c_j^y - c_i^y - m_\uparrow)\big) \quad R_{\text{left}}(p_i, p_j) = \text{sigmoid}\big(\kappa_\leftarrow(c_j^x - c_i^x - m_\leftarrow)\big)$$

$$R_{\text{h-align}}(p_i, p_j) = \exp\left(-\frac{(c_i^y - c_j^y)^2}{2\tau_h^2}\right) \qquad R_{\text{v-align}}(p_i, p_j) = \exp\left(-\frac{(c_i^x - c_j^x)^2}{2\tau_v^2}\right)$$

$$R_{\text{near}}(p_i, p_j) = \exp\left(-\frac{(c_i^x - c_j^x)^2 + (c_i^y - c_j^y)^2}{2\rho^2}\right)$$

$$R_{\text{contains}}(p_i, p_j) = \text{sigmoid}\big(\kappa_\supset \min\big[b_j^{x_1} - b_i^{x_1},\, b_j^{y_1} - b_i^{y_1},\, b_i^{x_2} - b_j^{x_2},\, b_i^{y_2} - b_j^{y_2}\big]\big).$$

Here, R_{above} and R_{left} score directional relations, indicating whether p_i is above or to the left of p_j in the image, respectively. The alignment predicates, $R_{\text{h}-\text{align}}$ and $R_{\text{v}-\text{align}}$ score whether two primitives share approximately the same vertical or horizontal position, while R_{near} scores spatial proximity. The containment predicate scores whether the soft bounding box of p_i encloses that of p_j. The learnable parameters κ control predicate sharpness, m denotes directional margins, and τ_h, τ_v, ρ control tolerance for alignment and proximity.

While binary predicates capture local spatial relations, object and part configurations often depend on geometric patterns involving multiple parts, such as angles, triangular arrangements, and relative distances between relations themselves (i.e., higher-order relations) (Biederman, 1987; Felzenszwalb et al., 2009). While it is desirable to model these relations, modeling higher-order relations and predicates exactly is difficult to learn and is also difficult to optimize directly. Thus, we express such patterns through enumerable ternary predicates. This keeps the structural layer differentiable and tractable, while allowing it to approximate higher-order layout cues over triples of primitives and pairs of primitive pairs. We describe this in detail below.

Ternary predicates. We model three-primitive geometric cues with two ternary predicates, R_{tri} and R_{turn}. For primitives p_i, p_j, p_k, let α_{ijk} denote the interior angle at p_i, and let $\theta^{\text{turn}}_{ijk} = \arccos(\hat{\mathbf{v}}_{ij} \cdot \hat{\mathbf{v}}_{jk})$ denote the turning angle along the ordered chain $p_i \to p_j \to p_k$. We define

$$R^{(n)}_{\text{tri}}(p_i, p_j, p_k) = \exp\left(-\frac{(\alpha_{ijk} - \psi_n)^2}{2\beta_n^2}\right) \quad R^{(\ell)}_{\text{turn}}(p_i, p_j, p_k) = \exp\left(-\frac{(\theta^{\text{turn}}_{ijk} - \phi_\ell)^2}{2\eta_\ell^2}\right).$$

Here, $R^{(n)}_{\text{tri}}$ scores triangular configurations with target angle ψ_n, while $R^{(\ell)}_{\text{turn}}$ scores ordered chain turns with learnable target angle ϕ_ℓ; β_n and η_ℓ control the corresponding tolerances.

Class-specific structural compositions. Together, these predicates define a differentiable vocabulary for spatial structure. While predicate parameters are shared globally, each class learns a sparse weighting over the enumerated predicate–primitive assignments. Given an image, CRISP evaluates these assignments on the detected primitives and aggregates their weighted scores into class evidence. Thus, classification depends jointly on primitive presence and on whether the detected primitives satisfy class-specific spatial patterns.

3.2 Network Architecture

In this section, we describe the network architecture that allows modeling and learning both the aforementioned primitives and structures. We give an overview of this architecture in Figure 1. This architecture consists of three main components: a visual backbone that allows extracting general visual features, a concept bottleneck layer that maps these features to the corresponding primitive along with their heatmaps and differential spatial coordinates as required by Section 3.1. Since the unary appearance predicate also requires the corresponding primitive features, we describe below how we use the region-of-interest (ROI) extractor to get these corresponding features. Having all the predicates and primitive features, we show how our structural scoring layer combines both these two sources of information to make predictions. We describe each of these components in turn below.

Visual backbone. Given an image $X \in \mathbb{R}^{3\times H\times W}$, we first use a visual backbone f_θ to extract a feature map $F = f_\theta(X) \in \mathbb{R}^{C\times H_F\times W_F}$, where C, H_F, W_F denote the channel, height, and width dimensions of the feature map.

Primitive concept bottleneck. This module instantiates the primitive descriptors defined in Section 3.1. Inspired by concept bottleneck models (Koh et al., 2020) and differentiable heatmap-based localization (Nibali et al., 2018), we project the feature map F into K primitive heatmaps,

$$F_H = g_{\text{cb}}(F) \in \mathbb{R}^{K\times H_F\times W_F},$$

where g_{cb} is a lightweight convolutional projection. For image X, the heatmap $F_H^{(k)}$ is the kth channel of F_H that provide spatial evidence for primitive p_k and is converted into the descriptor $z_k(X) = \langle c_k(X), \sigma_k(X), \delta_k(X), \ell_k(X)\rangle$. Following differentiable heatmap-to-coordinate transforms (Nibali et al., 2018), we form a temperature-normalized heatmap

$$\widetilde{F}_H^{(k)}[h,w] = \frac{\exp(F_H^{(k)}[h,w]/T)}{\sum_{h',w'}\exp(F_H^{(k)}[h',w']/T)}, \quad c_k(X) = \sum_{h,w}\widetilde{F}_H^{(k)}\odot G_{\text{coord}}, \quad \ell_k(X) = \sum_{h,w}\widetilde{F}_H^{(k)}\odot F$$

where T is a learnable temperature, $\odot$ is element-wise multiplication and $G_{\text{coord}} = \left\langle \frac{2w}{W_F-1} - 1, \frac{2h}{H_F-1} - 1 \right\rangle \in [-1,1]^2$ is the normalized grid coordinate map. We compute the presence confidence as $\sigma_k(X) = \text{sigmoid}(\max F_H^{(k)})$, and define $\delta_k(X)$ as twice the coordinate-wise standard deviation under $\widetilde{H}_k$. Intuitively, $\ell_k(X)$ is an attention-pooled appearance descriptor that aggregates backbone features from the regions emphasized by the primitive's heatmap. These descriptors provide the locations, confidences, and extents used by the structural scoring layer.

Localized scoring layer. While spatial predicates provide a general mechanism for composing visual primitives, class identity can additionally depend on local appearance. We therefore complement the structural scoring layer with a parallel appearance voting head that aggregates direct class evidence from individual primitive regions. For each primitive p_k, a shared detection-style ROI head f_{ROI} (He et al., 2017) pools backbone features F within its predicted bounding box $b_k(X)$ to a fixed 3×3 grid, averages them spatially, and maps the resulting vector to class logits. We aggregate these regional votes using normalized presence confidences:

$$\ell_{\text{ROI}}(X) = \sum_{k=1}^{K}\frac{\sigma_k(X)}{\sum_{j=1}^{K}\sigma_j(X)}\, f_{\text{ROI}}(F, b_k(X)) \in \mathbb{R}^{|\mathcal{C}|}.$$

Both branches operate on the same primitive set: the appearance head produces regional class votes from features pooled within predicted boxes, while the structural layer composes spatial predicates with appearance predicates derived from heatmap-pooled features.

Structural scoring layer. The structural scoring layer converts the structural vocabulary from Section 3.1 into class-level evidence. Given primitive descriptors $\{z_k(X)\}_{k=1}^{K}$, it evaluates unary presence scores and enumerates spatial predicate applications over valid primitive assignments:

$$a_{R,\mathbf{i}}(X) = R(p_{i_1}, \ldots, p_{i_r}; X), \quad R \in \mathcal{R}_r,\ \mathbf{i} = \langle i_1, \ldots, i_r \rangle \in \mathcal{A}_r.$$

where $\mathcal{R}_r$ denotes the set of r-ary predicates and $\mathcal{A}_r \subseteq \{1, \ldots, K\}^r$ contains the valid ordered assignments of primitives to their arguments. Collecting these activations yields $a(X) \in [0,1]^M$, where M is the total number of predicate applications. Ternary predicates account for most applications in our configuration, with an enumeration cost of $O(|\mathcal{R}_3|K^3)$. With $K = 16$, the layer evaluates approximately $8{,}000$ predicate applications per image. We parameterize class-specific structural compositions using $\Lambda = [\lambda^1; \lambda^2; \ldots; \lambda^{|\mathcal{C}|}] \in \mathbb{R}^{|\mathcal{C}| \times M}$, where $\lambda^c \in \mathbb{R}^M$ assigns an unnormalized weight to each predicate activation for class c. Unlike the globally shared predicate-shape parameters, these structural weights are class-specific. We normalize them with sparsemax (Martins and Astudillo, 2016) and compute the class-wise structural score $\ell_{\text{struct}}(X) \in [0,1]^{\mathcal{C}}$ as

$$w^c = \text{sparsemax}(\lambda^c) \in \Delta^{M-1}, \qquad \ell_{\text{struct}}(X)[c] = (w^c)^\top a(X)$$

Where Δ^{M-1} denotes the probability simplex. Thus, each class selects a sparse subset of primitive-presence and spatial-predicate activations as its structural composition. Because these activations are differentiable, gradients from the classification loss propagate through the structural scoring layer to the primitive bottleneck and visual backbone.

Class scoring and learning objective We compute the final class scores by summing the structural and appearance scores with log-softmax for numerical stability, $s(X) = \text{logsoftmax}(\ell_{\text{struct}}(X) + \ell_{\text{ROI}}(X)) \in \mathbb{R}^{|\mathcal{C}|}$ and train CRISP end-to-end with:

$$\mathcal{L} = \mathcal{L}_{\text{CE}}(s(X), y) + \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{DANN}}, \tag{1}$$

where $\mathcal{L}_{\text{CE}}$ is cross-entropy, $\mathcal{L}_{\text{reg}}$ stabilizes predicate learning and structural selection, and $\mathcal{L}_{\text{DANN}}$ helps to make the backbone features produced by f_θ domain-invariant, following (Ganin et al., 2016). As the architecture is fully differentiable, this objective jointly optimizes all the aforementioned components. We defines $\mathcal{L}_{\text{reg}}$ and $\mathcal{L}_{\text{DANN}}$ in details in Appendix B.

4 Experiments

We employ the widely used DomainBed (Gulrajani and Lopez-Paz, 2020; Cha et al., 2021) protocols and the challenging new CUB-DG dataset (Min et al., 2022) for evaluating our method. The rest of the dataset description and the implementation details can be found in our Appendix A. In the next sections, we describe our experimental results.

4.1 Overall Results

We report the mean accuracy of each of the 5 datasets in the DomainBed setting, and the CUB-DG dataset in Table 1. As can be seen in the table, our method is compatible with different backbone types such as the traditional ResNet-50 and the modern ViT-B/14 backbone in the DomainBed setting. Specifically, CRISP surpasses the current SOTA method ERM++ using both of the backbones with 1.2% and 0.4% on the DomainBed setting. On CUB-DG, CRISP achieves the best mean accuracy of 69.0%, outperforming the strongest prior result, ERM++, by 7.9 points. These results suggest that explicitly modeling spatial compositions is particularly effective on CUB-DG, where fine-grained bird categories share coarse part layouts while appearance varies substantially across domains.

4.2 Ablation Studies

We pose the following research questions (RQ)s as ablation studies for CRISP:

RQ1: What do the learned primitives capture? As illustrated in Fig. 3, CRISP's learned primitive 0 localizes class-relevant bird regions across source domains, such as the *wing-chest* region for class 4, the *neck* region for class 6, and the *wing-only* region for class 17. These localized primitives provide the building blocks for CRISP to model class-specific spatial relations. Furthermore, their localization patterns remain relatively consistent across changes in texture and pose, suggesting that

Table 1: Accuracy on DomainBed and CUB-DG. Best in **bold**; second-best in underlined.

	Method	PACS	VLCS	OfficeHome	TerraInc	DomainNet	CUB-DG	DomainBed
ResNet-50	ERM	84.2	77.3	67.6	47.8	44.0	-	64.2
	DANN	83.6	78.6	65.9	46.7	38.3	49.5	62.6
	CDANN	82.6	77.5	65.8	45.8	38.3	48.6	62.0
	GroupDRO	84.4	76.7	66.0	43.2	33.3	44.8	60.7
	MixUp	85.2	77.9	60.4	44.0	34.0	50.3	60.3
	SagNet	86.3	77.8	68.1	48.6	40.3	51.6	64.2
	CORAL	86.2	78.8	68.7	47.7	41.5	55.5	64.6
	GVRT	85.1	79.0	70.1	48.0	44.1	57.0	65.3
	MIRO	85.4	79.0	70.5	50.4	44.3	49.1	65.9
	SWAD	88.1	79.1	70.6	50.0	46.5	52.2	66.9
	ERM++	**89.8**	78.0	**74.7**	51.2	**50.8**	61.1	68.9
	CRISP (Ours)	88.4	**80.4**	73.4	**58.5**	50.0	**69.0**	**70.1**
ViT-B/14	ERM	83.4	75.9	66.4	35.3	44.4	-	61.1
	MIRO	95.6	82.5	82.5	54.3	54.0	-	73.7
	CAR-FT	96.8	85.5	85.7	61.9	**62.5**	-	78.5
	UNI-DG	96.7	**86.3**	86.2	62.4	61.3	-	78.6
	ERM++	96.8	82.4	86.7	**67.4**	59.8	**81.2**	78.6
	CRISP (Ours)	**97.0**	84.4	**88.0**	67.0	58.5	80.1	**79.0**

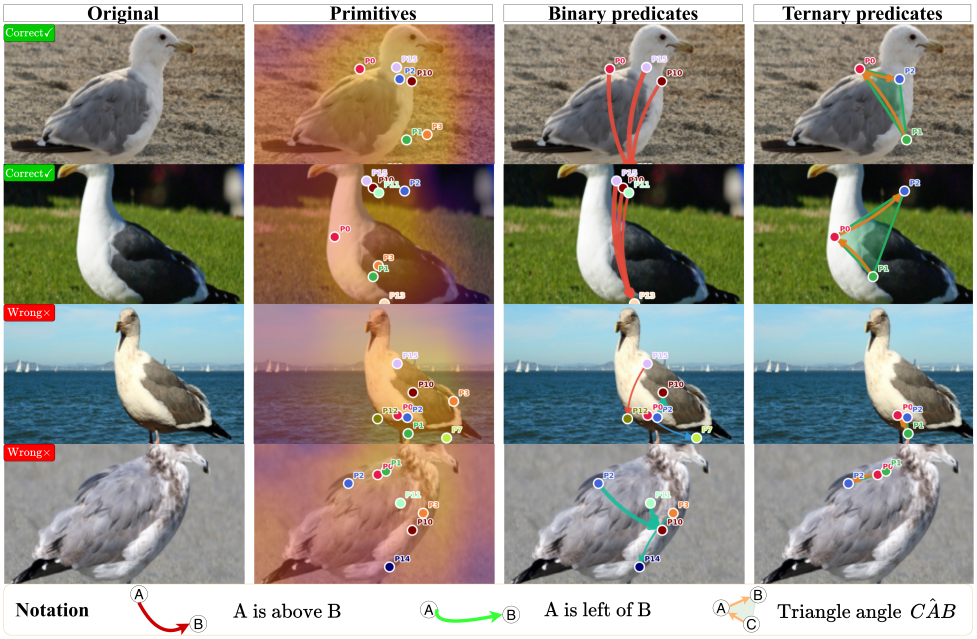

Figure 2: Error analysis on 2 correct and 2 false classification samples of the target (Photo) domain. Error happens due to the misidentification of the crucial primitive region P_{13}.

the learned primitives capture spatially meaningful evidence that is robust to domain shift. We will display more closely the per-dataset results and learned primitives with their corresponding relations in Appendix C.

RQ2: What failure modes does CRISP exhibit? Fig. 2 shows that primitive localization is a key bottleneck for CRISP. In correctly classified samples, primitives localize informative regions: for example, P_{15} and P_{10} focus on the neck–head region, while P_{13} captures the feet. In misclassified samples, several primitives collapse around the torso and fail to anchor to class-discriminative extremities, including the region typically captured by P_{13}. This weakens the geometric evidence available to the structural scoring layer, making the final prediction more error-prone.

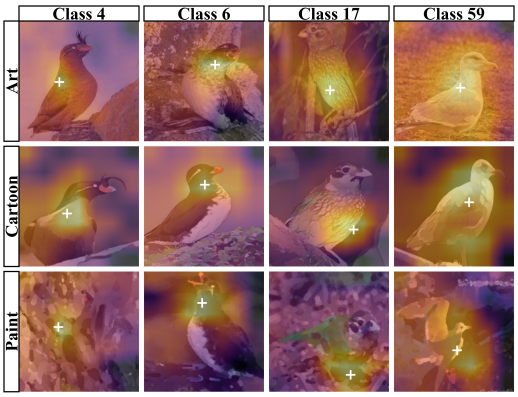

Figure 3: Primitive 0 visualization over different classes of source domains.

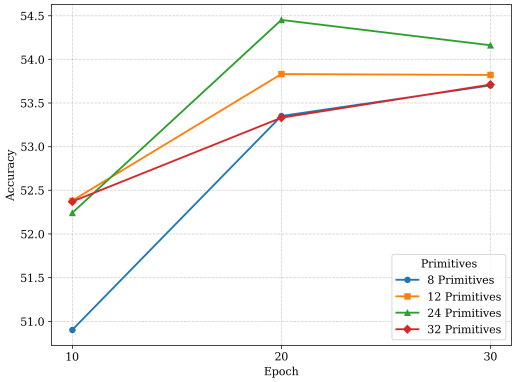

Figure 4: Results of CRISP with different numbers of primitives on DomainNet/Sketch dataset.

Table 2: Contributions of network components.

	Method	PACS	VLCS	OH	Terra	DMN	Avg.
RN50	CRISP (ERM++)	**89.8**	78.4	73.1	57.8	48.8	69.6
	CRISP (w/o ROI)	86.7	76.9	70.5	54.8	44.4	66.7
	CRISP (w/o Grammar)	87.2	75.0	71.3	52.6	45.6	66.3
	CRISP (full)	88.4	**80.4**	**73.4**	**58.5**	**50.0**	**70.1**
ViT-B/14	CRISP (w/o Grammar)	94.8	82.5	85.9	63.9	57.8	77.0
	CRISP (full)	**97.0**	**84.4**	**88.0**	**67.0**	**58.5**	**79.0**

Table 3: Performance of predicate sets.

	Art	Cartoon	Paint	Photo	Avg.
No Rel.	66.3	62.0	45.8	68.6	60.7
Binary	69.6	68.7	47.9	75.3	65.4
Binary + Ternary	71.7	71.5	50.5	76.8	67.6
All	72.4	72.3	51.9	79.6	69.0

RQ3: How does the number of primitives affect performance? We study the effect of the number of primitives on the DomainNet task with Sketch as the target domain, as shown in Fig. 4. Increasing the number of primitives from the default setting to $K = 24$ improves accuracy by 0.5 points, suggesting that additional primitives can provide finer spatial evidence. However, since the number of enumerated predicate applications grows with K, larger primitive sets increase the computational cost of the structural scoring layer. We therefore use the smaller default configuration for the DomainBed experiments as a trade-off between accuracy and efficiency.

RQ4: What is the contribution of each components in CRISP? We dedicate Table 2 and Table 3 to study the contributions of different training strategies and network components. Specifically, Table 2 studies whether applying ERM++ Teterwak et al. (2024) (CRISP (ERM++)) would yield better performance. We also study the contribution of a localized scoring layer (through CRISP (w/o ROI)) and that of the structural scoring layer (through CRISP w/o ROI)), respectively. The results show that the full CRISP architecture, with both localized and structural scoring layers, outperforms its ablated baselines consistently, and on both convolutional and transformer-based backbones. Interestingly, CRISP (ERM++) gives the new state-of-the-art on the PACS dataset with the RN50 backbone. We also perform more fine-grained ablations on the predicate sets on CUB-DG in Table 3 and show that adding binary, ternary, their combination, and their combinations with unary appearance predicates consistently improve CRISP's overall performance.

5 Conclusion

We presented CRISP, an end-to-end domain generalization framework that combines learned visual primitives with differentiable spatial and appearance predicates. Class-specific sparse compositions of these predicates provide structural evidence, complemented by an appearance voting head operating on the same primitive regions. Across five DomainBed datasets, CRISP achieves mean accuracies of 70.1% with ResNet-50 and 79.0% with ViT-B/14, improving over ERM++ by 1.2 and 0.4 percentage points, respectively. On CUB-DG with ResNet-50, it achieves 69.0% mean accuracy, exceeding ERM++ by 7.9 percentage points. Predicate ablations show that binary and ternary spatial relations improve over primitive-only classification, while shared unary appearance predicates provide further gains. These findings support combining explicit relational composition with local appearance evidence for domain generalization.

REFERENCES

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

Shuanghao Bai, Min Zhang, Wanqi Zhou, Siteng Huang, Zhirong Luan, Donglin Wang, and Badong Chen. Prompt-based distribution alignment for unsupervised domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 729–737, 2024.

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

Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987.

Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34:22405–22418, 2021.

Junbum Cha, Kyungjae Lee, Sungrae Park, and Sanghyuk Chun. Domain generalization by mutual-information regularization with pre-trained models. In *European Conference on Computer Vision (ECCV)*, pages 440–457. Springer, 2022. doi: 10.1007/978-3-031-20050-2_26.

Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020.

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*, pages 248–255. Ieee, 2009.

Chen Fang, Ye Xu, and Daniel N Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE international conference on computer vision*, pages 1657–1664, 2013.

Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016. URL `http://jmlr.org/papers/v17/15-239.html`.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2020.

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*, pages 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

Joy Hsu, Jiayuan Mao, Josh Tenenbaum, and Jiajun Wu. What's left? concept grounding with logic-enhanced foundation models. *Advances in Neural Information Processing Systems*, 36: 38798–38814, 2023.

Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017.

Danial Kamali, Elham J. Barezi, and Parisa Kordjamshidi. Nesycoco: A neuro-symbolic concept composer for compositional generalization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(4):4184–4193, Apr. 2025. doi: 10.1609/aaai.v39i4.32439. URL https://ojs.aaai.org/index.php/AAAI/article/view/32439.

Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19113–19122, June 2023.

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

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5338–5348. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/koh20a.html.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

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*, pages 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*, pages 5542–5550, 2017.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C. Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Runtao Liu, Chenxi Liu, Yutong Bai, and Alan L Yuille. Clevr-ref+: Diagnosing visual reasoning with referring expressions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4185–4194, 2019.

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

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018. doi: 10.48550/arxiv.1705.10667.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv preprint arXiv:1904.12584*, 2019.

Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International conference on machine learning*, pages 1614–1623. PMLR, 2016.

Seonwoo Min, Nokyung Park, Siwon Kim, Seunghyun Park, and Jinkyu Kim. Grounding visual representations with texts for domain generalization. In *European Conference on Computer Vision (ECCV)*, pages 37–53. Springer, 2022. doi: 10.1007/978-3-031-19836-6_3.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *ICML (1)*, pages 10–18, 2013.

Hyeonseob Nam, HyunJae Lee, JongChan Park, Wonjun Cho, and Kwanyoung Kim. Reducing domain gap by reducing style bias. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8690–8699, 2021. doi: 10.1109/CVPR46437.2021.00858.

Duc-Duy Nguyen and Dat Nguyen. VirDA: Reusing backbone for unsupervised domain adaptation with visual reprogramming. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=Qh7or7JRFI.

Aiden Nibali, Zhen He, Stuart Morgan, and Luke Prendergast. Numerical coordinate regression with convolutional neural networks. *arXiv preprint arXiv:1801.07372*, 2018.

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*, pages 1406–1415, 2019.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

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. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=ryxGuAnKwH.

Yang Shu, Zhangjie Cao, Chenyu Wang, Jianmin Wang, and Mingsheng Long. Open domain generalization with domain-augmented meta-learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9624–9633, 2021.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision (ECCV)*, pages 443–450. Springer, 2016. doi: 10.1007/978-3-319-49409-8_35.

Tao Sun, Cheng Lu, Tianshuo Zhang, and Haibin Ling. Safe self-refinement for transformer-based domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Piotr Teterwak, Kuniaki Saito, Theodoros Tsiligkaridis, Kate Saenko, and Bryan A Plummer. Erm++: An improved baseline for domain generalization. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024. doi: 10.48550/arxiv.2304.01973.

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*, pages 5018–5027, 2017.

Tongkun Xu, Weihua Chen, Pichao Wang, Fan Wang, Hao Li, and Rong Jin. Cdtrans: Cross-domain transformer for unsupervised domain adaptation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. ICLR 2022 (poster). URL: https://openreview.net/forum?id=XGzk5OKWFFc.

Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B. Tenenbaum. Neural-symbolic vqa: disentangling reasoning from vision and language understanding. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 1039–1050, Red Hook, NY, USA, 2018. Curran Associates Inc.

Hanlin Zhang, Yi-Fan Zhang, Weiyang Liu, Adrian Weller, Bernhard Schölkopf, and Eric P Xing. Towards principled disentanglement for domain generalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8024–8034, 2022.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018. doi: 10.48550/arxiv.1710.09412.

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. *arXiv preprint arXiv:2104.02008*, 2021.

APPENDIX

A DATASET AND IMPLEMENTATION DETAILS

The DomainBed consists of 5 datasets: PACS (Li et al., 2017) (4 domains, 7 classes, and 9,991 images), VLCS (Fang et al., 2013) (4 domains, 5 classes, and 10,729 images), Office-Home (Venkateswara et al., 2017) (4 domains, 65 classes, and 15,588 images), TerraIncognita (Beery et al., 2018) (4 domains, 10 classes, and 24, 788 images), and DomainNet (Peng et al., 2019) (6 domains, 345 classes, and 586,575 images). The CUB-DG dataset comprises 4 domains, with 47,152 images across 200 classes of North American bird species. All performance scores are evaluated by *leave-one-out* cross-validation, where averaging all cases that use a single domain as the target (test) domain and the others as the source (training) domains. We hold out 20% of the source domain data for validation and select the best validation model for testing on the held-out target domain. Since we use ResNet-50 (He et al., 2016) backbone pretrained on ImageNet (Deng et al., 2009) as backbone, we benchmark against prior methods that utilize the same backbone architecture for fair comparison, which are: DANN (Ganin et al., 2016), CDANN (Long et al., 2018), GroupDRO (Sagawa et al., 2020), MixUp (Zhang et al., 2018), SagNet (Nam et al., 2021), CORAL (Sun and Saenko, 2016), ERM++ (Teterwak et al., 2024), MIRO (Cha et al., 2022), and SWAD (Cha et al., 2021). We also include GVRT (Min et al., 2022) as the current benchmark due to the remarkable SoTA performance on the CUB-DG dataset.

Implementation details. The model is optimized using the Adam (Kingma and Ba, 2014) optimizer with a default learning rate of 0.001. We set the number of primitives to $K = 16$ for all domains in the DomainBed and CUB-DG settings, while keeping the same batch size $N = 32$ to be consistent with prior methods. The number of target angles n and turning angles ℓ of ternary predicates are set to 3 and 1, respectively. Following prior DG work, we apply feature-level style mixing (Zhou et al., 2021) during training to reduce sensitivity to domain-specific texture statistics. We achieve these hyperparameters via grid search, which is reproducible via the source code. All of the experiments were conducted on a single RunPod cloud instance equipped with 9 vCPUs, 50 GB system RAM, and 60 GB of storage, and an NVIDIA RTX 4000 Ada Generation GPU with 20 GB GDDR6 VRAM. For the training time, it takes 0.8 seconds per step, resulting in approximately 440s per epoch on the CUB-DG dataset.

B REGULARIZATION AND DOMAIN ADVERSARIAL LOSS

This section provides the full definition of the regularization and the adversarial term used in Eq. 1. The full training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}}(s(X), y) + \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{DANN}}$$

where

$$\mathcal{L}_{\text{reg}} = \lambda_{\text{sparse}} \overline{\|\Lambda\|}_1 + \lambda_{\text{bn}} \mathcal{L}_{\text{bn}} + \lambda_{\text{ang}} \mathcal{L}_{\text{ang}},$$

where $\mathcal{L}_{\mathrm{CE}}$ is the standard cross-entropy loss over structural class scores, and the remaining terms are described below. The predicate parameters receive no direct supervision beyond $\mathcal{L}_{\mathrm{CE}}$; they are optimized entirely through the gradient path from class scores through grammar weights to relation scores.

Sparsity regularization. The class-weight matrix $\Lambda \in \mathbb{R}^{|\mathcal{C}| \times M}$ holds class-specific log-weights over the M enumerated predicate applications. We apply a mean ℓ_1 penalty,

$$\overline{\|\Lambda\|}_1 = \frac{1}{|\mathcal{C}|\, M} \sum_{c=1}^{|\mathcal{C}|} \sum_{m=1}^{M} |\Lambda_{c,m}|,$$

which complements sparsemax normalization by penalizing log-weight magnitude before the activation, encouraging each class to rely on a compact subset of predicate applications.

Bottleneck regularization. To encourage the concept bottleneck to learn spatially meaningful and non-redundant primitive detectors, we regularize the normalized primitive heatmaps $\{\widetilde{H}_k\}$. We define

$$\mathcal{L}_{\text{bn}} = \mathcal{L}_{\text{div}} + \lambda_{\text{conc}} \mathcal{L}_{\text{conc}},$$

Table 4: Accuracy on PACS. Best in **bold**; second-best underlined, within each backbone.

Method	Art painting	Cartoon	Photo	Sketch	Avg.
ResNet-50 Backbone					
ERM	84.7	80.8	97.2	79.3	84.2
ERM + SWAD	89.3	83.4	97.3	82.5	88.1
DIWA	90.6	83.4	98.2	83.8	89.0
ERM + MIRO + SWAD	–	–	–	–	88.4
ERM++	90.6	83.7	98.1	**86.6**	**89.8**
ERM++ + MIRO	90.2	**83.8**	98.6	82.4	88.8
CRISP (Ours)	**91.9**	83.2	**98.8**	79.5	88.4
ViT-B/14 Backbone					
ERM++	–	–	–	–	96.8
CRISP (Ours)	**99.1**	**96.8**	**100.0**	**92.1**	**97.0**

where $\mathcal{L}_{\text{div}}$ discourages different primitives from attending to the same spatial regions, and $\mathcal{L}_{\text{conc}}$ encourages each primitive heatmap to be spatially concentrated. Concretely, flattening each normalized heatmap $\widetilde{H}_k$ into a vector in $\Delta^{H_F W_F - 1}$, we compute

$$\mathcal{L}_{\text{div}} = \frac{1}{K(K-1)} \sum_{k \neq k'} \frac{\widetilde{H}_k^\top \widetilde{H}_{k'}}{\|\widetilde{H}_k\|_2 \, \|\widetilde{H}_{k'}\|_2}, \quad \mathcal{L}_{\text{conc}} = -\frac{1}{K} \sum_{k=1}^{K} \sum_{h,w} \widetilde{H}_{k,h,w} \log(\widetilde{H}_{k,h,w} + \epsilon).$$

Where $\epsilon = 0.01$ is the normalization term. This bottleneck regularizer is used in every training run and helps reduce primitive collapse.

Domain Adversarial Features regularization. For each pair of (X, d), denote the input image with the corresponding source domain label; the goal of this regularization term is to make the extracted features F from the backbone f_θ distributionally indistinguishable. Thus, following the work of (Ganin et al., 2016), we employ a Gradient Reversal Layer (GRL) with a domain classifier f_{Dom}. In practice, f_{Dom} is implemented as a Linear classifier that predicts which domain the features come from:

$$f_{Dom}(F) = \hat{d} \in \mathbb{R}^{|\mathcal{S}|}$$

where $|\mathcal{S}|$ is the total number of source domains. To achieve domain confusion, we apply a standard cross-entropy loss $\mathcal{L}_{\text{DANN}} = \mathcal{L}_{\text{CE}}(\hat{d}, d)$ to the predictions of f_{Dom}. During training, the GRL acts as an identity transformation in the forward pass. However, during the backward pass, it multiplies the gradient from $\mathcal{L}_{\text{DANN}}$ by a negative scalar $-\lambda$. This forces the domain classifier f_{Dom} to minimize the classification error while simultaneously driving the backbone f_θ to maximize it, ultimately resulting in domain-invariant feature representations.

C Per-Dataset results

Results on PACS Table 4 reports per-domain accuracy on PACS. With ResNet-50, CRISP achieves the highest accuracy among the listed methods on Art painting and Photo, while its Sketch accuracy falls sharply to 7.1 percentage points compared with ERM++, resulting in a lower average of 88.4% versus 89.8%. With ViT-B/14, CRISP reaches 97.0% average accuracy, slightly exceeding ERM++ by 0.2 percentage points. These results indicate that improvements are domain-dependent, particularly when transferring to sketches with the ResNet-50 backbone. As illustrated in Fig. 5, the correctly classified elephant sketches retain identifiable head and trunk regions whose spatial arrangement resembles that highlighted in the source examples despite substantial appearance changes. In the failed example predicted as a giraffe, the highlighted regions provide a less distinct head and trunk structure, suggesting that ambiguous primitive localization can weaken the relational evidence for classification.

Results on VLCS As illustrated in Table 5, CRISP with ResNet-50 achieves the highest average accuracy of 80.4%, outperforming ERM++ by 2.4 percentage points and the strongest listed baseline,



Figure 5: The visualization of samples from all 6 datasets: CUB-DG, VLCS, PACS, OfficeHome, TerraIncognita, and DomainNet. For each dataset, we visualize a class with the corresponding source samples, as well as two correctly and one incorrectly classified samples from the target domain. Furthermore, each image visualizes the most prominent production, with its primitives and relation, denoted with different centroids and bounding boxes.

Table 5: Accuracy on VLCS. Best in **bold**; second-best underlined, within each backbone.

Method	Caltech101	LabelMe	SUN09	VOC2007	Avg.
ResNet-50 Backbone					
ERM	97.7	64.3	73.4	74.6	77.3
ERM + SWAD	98.8	63.3	**75.3**	79.2	79.1
DIWA	98.9	62.4	73.9	78.9	78.6
ERM + MIRO + SWAD	–	–	–	–	79.6
ERM++	98.7	63.2	71.6	78.7	78.0
ERM++ + MIRO	99.0	62.4	71.8	78.3	77.9
CRISP (Ours)	**99.7**	**67.2**	**75.3**	**79.4**	**80.4**
ViT-B/14 Backbone					
ERM++	–	–	–	–	82.4
CRISP (Ours)	**99.6**	**69.6**	**81.6**	**86.8**	**84.4**

Table 6: Accuracy on OfficeHome. Best in **bold**; second-best underlined, within each backbone.

Method	Art	Clipart	Product	Real	Avg.
ResNet-50 Backbone					
ERM	63.1	51.9	77.2	78.1	67.6
ERM + SWAD	66.1	57.7	78.4	80.2	70.6
DIWA	69.2	59.0	81.7	82.2	73.0
ERM + MIRO + SWAD	–	–	–	–	72.4
ERM++	70.7	**62.2**	81.8	84.0	74.7
ERM++ + MIRO	**74.0**	61.5	**83.8**	**85.7**	**76.3**
CRISP (Ours)	71.7	57.7	81.1	83.0	73.4
ViT-B/14 Backbone					
ERM++	–	–	–	–	86.7
CRISP (Ours)	**88.0**	**79.6**	**91.4**	**92.9**	**88.0**

ERM + MIRO + SWAD, by 0.8 points in the VLCS dataset. Relative to ERM++, the largest improvements occur on LabelMe and SUN09, with gains of 4.0 and 3.7 points, respectively. With ViT-B/14, CRISP improves average accuracy from 82.4% to 84.4%, showing gains over ERM++ with both backbones. In Fig. 5, the successful target examples show that the highlighted regions and their relations can support bird recognition even when the birds occupy a small portion of the scene. In the failed example predicted as a chair, the highlighted regions lie largely on architecture and surrounding scenery, suggesting that background structure can introduce misleading relational cues.

Results on OfficeHome Table 6 compares per-domain top-1 accuracy on OfficeHome. With ResNet-50, CRISP achieves 73.4% average accuracy, exceeding ERM, ERM + SWAD, DIWA, and ERM + MIRO + SWAD, but remaining below ERM++ and ERM++ + MIRO. Although it improves Art accuracy over ERM++ by 1.0 percentage point, its Clipart accuracy decreases by 4.5 points, contributing to an overall reduction of 1.3 points. With ViT-B/14, CRISP achieves the highest reported average of 88.0%, outperforming ERM++ by 1.3 points. The different outcomes across backbones suggest that the effectiveness of the primitive-based representation depends partly on the underlying visual features. As shown in Fig. 5, the correctly classified calculator examples preserve recognizable keypad regions and their spatial arrangement despite changes in rendering and viewpoint. The failed example predicted as a calendar also contains a regular grid of rectangular elements, illustrating how similar local primitives and layouts can create ambiguity between semantically different categories.

Results on Terra Incognita **Terra Incognita.** Table 7 reports accuracy across the four held-out locations of Terra Incognita. With ResNet-50, CRISP achieves the highest average accuracy of 58.5%, improving over ERM++ by 7.3 percentage points and ERM++ + MIRO by 5.1 points. It

Table 7: Accuracy on Terra Incognita. Best in **bold**; second-best underlined, within each backbone.

Method	Loc. 100	Loc. 38	Loc. 43	Loc. 46	Avg.
ResNet-50 Backbone					
ERM	54.3	42.5	55.6	38.8	47.8
ERM + SWAD	55.4	44.9	59.7	39.9	50.0
DIWA	57.2	50.1	60.3	39.8	51.9
ERM + MIRO + SWAD	–	–	–	–	52.9
ERM++	48.3	**50.7**	61.8	43.9	51.2
ERM++ + MIRO	60.8	48.8	61.1	42.7	53.4
CRISP (Ours)	**70.7**	49.2	**64.7**	**49.2**	**58.5**
ViT-B/14 Backbone					
ERM++	–	–	–	–	**67.4**
CRISP (Ours)	**77.1**	**56.3**	**68.8**	**65.9**	67.0

Table 8: Accuracy on DomainNet. Best in **bold**; second-best underlined, within each backbone.

Method	Painting	Clipart	Info	Real	Quickdraw	Sketch	Avg.
ResNet-50 Backbone							
ERM	50.1	63.0	21.2	63.7	13.9	52.9	44.1
ERM + SWAD	53.5	66.0	22.4	65.8	16.1	55.5	46.6
DIWA	55.4	66.2	23.3	68.7	16.5	56.0	47.7
ERM + MIRO + SWAD	–	–	–	–	–	–	47.0
ERM++	58.4	**71.5**	26.2	70.7	**17.3**	**60.5**	**50.8**
ERM++ + MIRO	58.5	71.0	26.5	**71.1**	15.9	59.5	50.4
CRISP (Ours)	**59.9**	69.0	**28.2**	70.7	14.3	57.6	50.0
ViT-B/14 Backbone							
ERM++	–	–	–	–	–	–	**59.8**
CRISP (Ours)	**69.6**	**76.8**	**39.0**	**79.9**	**18.6**	**67.0**	58.5

obtains the best results at Locations 100, 43, and 46, with the largest improvement over ERM++ occurring at Location 100: 70.7% versus 48.3%. Performance at Location 38 remains below ERM++, indicating that the gains are not uniform across environments. With ViT-B/14, CRISP achieves 67.0%, slightly below the 67.4% obtained by ERM++. In Fig. 5, the correctly classified coyote examples retain a spatial arrangement between a broad scene region and a localized animal region resembling that highlighted in the source images, despite changes in illumination and background. In the failed example predicted as a raccoon, the highlighted regions shift toward a small background area, suggesting that inaccurate localization can replace useful animal-related evidence with scene-specific cues.

Results on DomainNet Table 8 presents per-domain top-1 accuracy on DomainNet. With ResNet-50, CRISP obtains the best results on Painting and Infograph, improving over ERM++ by 1.5 and 2.0 percentage points, respectively. However, these gains are offset by lower accuracy on Clipart, Quickdraw, and Sketch, resulting in an average of 50.0%, compared with 50.8% for ERM++. With ViT-B/14, CRISP reaches 58.5%, also below the 59.8% achieved by ERM++. Thus, although the method benefits some visual domains, the results do not demonstrate a consistent advantage across the diverse representations in DomainNet. As illustrated in Fig. 5, the correctly classified hedgehog sketches retain recognizable face and spiny-body regions whose relative arrangement remains visible across the source styles. In the failed example predicted as an owl, the highlighted regions span multiple owl drawings and repeated textures, suggesting that fragmented localization and superficially similar patterns can produce misleading primitive relations.

Table 9: Accuracy on CUB-DG. Best in **bold**; second-best underlined, within each backbone.

Method	Photo	Cartoon	Art	Paint	Avg.
ResNet-50 Backbone					
DANN	67.5	57.0	42.8	30.6	49.5
CDANN	65.3	55.2	43.2	30.5	48.6
GroupDRO	60.9	54.8	36.5	27.0	44.8
MixUp	67.1	55.9	51.1	27.2	50.3
SagNet	67.4	60.7	44.0	34.2	51.6
CORAL	72.2	63.5	50.3	35.8	55.5
GVRT (PTE)	74.6	64.2	52.2	37.0	57.0
MIRO	60.1	49.9	58.9	34.9	51.0
SWAD	64.0	54.7	52.1	37.9	52.2
ERM++	60.3	61.6	56.8	**65.8**	61.1
CRISP (Ours)	**79.6**	**72.3**	**72.4**	51.9	**69.1**
ViT-B/14 Backbone					
ERM++	**93.0**	82.4	83.7	**65.7**	**81.2**
CRISP (Ours)	89.0	**85.0**	**84.1**	62.3	80.1

Results on CUB-DG Table 9 reports per-domain top-1 accuracy on CUB-DG. With ResNet-50, CRISP achieves the highest average accuracy of 69.1%, exceeding the strongest listed baseline, ERM++, by 8.0 percentage points. It obtains the best results on Photo, Cartoon, and Art, outperforming the strongest competing result in each domain by 5.0, 8.1, and 13.5 points, respectively. However, its Paint accuracy is 13.9 points below ERM++, showing that these gains do not extend to every visual style. With ViT-B/14, CRISP improves Cartoon and Art accuracy over ERM++ but performs worse on Photo and Paint, yielding a lower average of 80.1% versus 81.2%. As demonstrated in Fig. 5, the correctly classified target birds retain highlighted primitives around the leg and tail regions, providing spatial cues that remain recognizable across changes in style and pose. In the failed example predicted as a bank swallow, the primitive regions include substantial background and perch information, which may obscure the relevant anatomical relations and contribute to the incorrect classification.