Brown University

Department of Computer Science

# Vision-Language Guided Evidence Alignment for Robust Classification: A Study of R4RR and Extensions

A thesis submitted in partial fulfillment of Honors in Computer Science

by

Ryan Yang

Providence, Rhode Island

May 2026

# Acknowledgements

I am deeply grateful to Dr. Dipkamal Bhusal (RIT), Professor Nidhi Rastogi (RIT), and Professor Daniel Ritchie (Brown University) for their guidance and support throughout this project.

# Abstract

Image classifiers often achieve high average accuracy by exploiting spurious cues such as background or texture, leading to brittle behavior under distribution shift. We propose **Right for the Right Regions (R4RR)**, a scalable framework that reduces shortcut reliance by aligning a classifier's spatial evidence with task-relevant *localization maps* produced by a frozen vision–language teacher. Instead of penalizing noisy input gradients or requiring human pixel masks, R4RR aligns *latent evidence*: we extract Class Activation Maps (CAMs) from the classifier and match them to the teacher maps using a Kullback–Leibler divergence between spatial probability distributions. We adopt a simple two-phase training protocol, *Classify-then-Align*, that warm-starts the classifier with standard cross-entropy before introducing evidence alignment, improving stability in the presence of competing objectives. Across three complementary benchmarks—DecoyMNIST, Waterbirds, and a RedMeat task derived from Food-101—R4RR consistently improves robustness and generalization over standard ERM training and representative baselines based on language-guided attention supervision, architectural attention-branch mechanisms, and automatic feature reweighting. We provide our code in the supplementary material.

**Keywords:** spurious correlations, vision–language models, distribution shift, robust classification

# Contents

**Acknowledgements** **i**

**Abstract** **ii**

**Main Text** **1**

1 Introduction . . . . . 1
2 Related Work . . . . . 3
3 Methodology . . . . . 5
3.1 Problem Setup and Notation . . . . . 5
3.2 Overview . . . . . 5
3.3 Teacher: VLM-Derived Semantic Affinity Maps . . . . . 5
3.4 Student Evidence: Class Activation Maps . . . . . 6
3.5 Evidence Alignment Objective . . . . . 7
3.6 Two-Phase Training: Classify-then-Align . . . . . 8
4 Experiments . . . . . 8
4.1 Experimental Setup . . . . . 8
4.2 Results . . . . . 11
4.3 Attention evaluation with the Pointing Game . . . . . 13
5 Ablation studies . . . . . 15
5.1 Study on training curriculum . . . . . 15
5.2 Study on teacher selection . . . . . 15
5.3 Ablation: teacher maps vs. objective/schedule . . . . . 16
5.4 Teacher-map corruption stress test . . . . . 17
6 Limitations . . . . . 18
7 Conclusion and Future Work . . . . . 18

**Appendix** **22**

A Prompts . . . . . 22

B Implementation Details . . . . . 23
B.1 Model Details by Dataset . . . . . 23
B.2 Teacher map storage . . . . . 24
B.3 Hyperparameter Optimization . . . . . 24
B.4 R4RR Optimized Hyperparameters . . . . . 27
C Local sensitivity to R4RR-specific hyperparameters . . . . . 28
D Teacher ablation on additional datasets . . . . . 29
E Dataset splits . . . . . 30
F Additional GALS Optimization: Exhaustive Variant Selection . . . . . 31

**Detailed Acknowledgements** **33**

**Bibliography** **33**

# Main Text

## 1 Introduction

Modern image classifiers achieve impressive average accuracy, yet often do so for the wrong reasons. When training labels correlate with background, texture, or acquisition artifacts, Convolutional Neural Networks (CNNs) can maximize training performance by relying on these shortcut cues rather than the intended object evidence [11]. Such behavior is largely invisible under standard i.i.d. evaluation but becomes apparent under distribution shift. The classifier fails when the spurious correlation breaks, revealing that its decision rule was never aligned with the task semantics [25]. A canonical example is the *Waterbirds* dataset [26], where bird type (landbird vs. waterbird) is spuriously correlated with background (land vs. water) during training, causing models to over-rely on background and fail on minority groups at test time.

Instead of only being *right*, we want models to be *right for the right regions*—their spatial evidence should align with the object rather than spurious context. As shown in Fig. 1, the CAM evidence should shift from background shortcuts to semantically relevant regions.

A natural response is to regularize explanations so models are *right for the right reasons*. Classical approaches such as Right for the Right Reasons (RRR) [25] and CDEP [24] penalize input-gradient or saliency signals on human-specified irrelevant regions, while new concept-based objectives steer representations toward human-interpretable concepts via concept-sensitive losses [12]. Despite their promise, these methods often face a *supervision bottleneck*: they rely on pixel-level masks, curated concept sets, or expert rationales that are expensive to obtain and may encode annotator subjectivity. This motivates using language as a scalable prior. A long line of work conditions vision models on textual descriptions or attributes [5, 7, 16, 17, 23], and more recent vision–language models can produce spatial cues (grounding, attention, affinity maps) from prompts without manual masks [19, 33]. Building on this idea for debiasing, GALS [20] derives saliency guidance from language specifications and enforces it via input-gradient regularization. However, input-centric supervision is noisy, and jointly optimizing classification and attention constraints from the start can create a

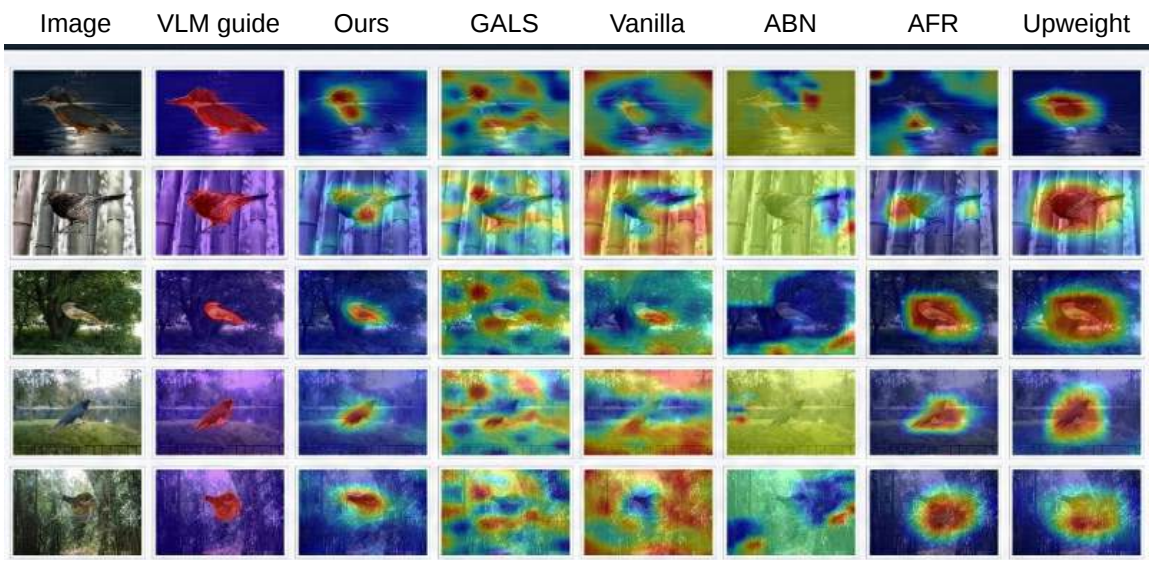


Figure 1: **Right for the wrong regions vs. right for the right regions.** Comparing saliency maps between our method, *GALS* [20], standard *vanilla* model, *ABN* [8], *AFR* [22] and class balanced model (*Upweight*). Shortcut-prone models often attend to spurious context (e.g., background).

competitive objective landscape in which shortcut features are learned early and become difficult to correct.

In this work, we propose **Right for the Right Regions (R4RR)**, a scalable framework for automated "right for the right reasons" training that treats VLMs as high-level teachers and shifts guidance from input gradients to *latent spatial evidence*. Concretely, we align the CNN's Class Activation Maps (CAMs) with dense semantic affinity maps produced by a strong CLIP-based localization teacher (WeCLIP+ [33]). Rather than enforcing agreement with a single salient hotspot, we match evidence at the level of coherent regions by casting both maps as spatial probability distributions and minimizing a Kullback–Leibler divergence. This yields a smooth distributional alignment signal that avoids brittle hard-masking and reduces reliance on pixel-gradient suppression.

A central component of **R4RR** is a simple yet effective *two-phase* optimization schedule designed to manage objective competition. In our main setting, we find that *Classify-then-Align* works best across datasets. We first train a model with cross-entropy to learn discriminative structure, then introduce evidence alignment and optimize the combined objective. This warm start reduces the tendency to lock onto shortcuts before alignment becomes effective. We include the reverse curriculum (*Align-then-Classify*) as an ablation study.

We evaluate **R4RR** on three complementary benchmarks of spurious correlation: *DecoyM-*

*NIST* [6], *Waterbirds* [26], and *RedMeat*, derived from Food-101 [2]. Across all three datasets, **R4RR** consistently outperforms language-guided attention baseline, *GALS* [20], attention-module baseline, *ABN* [8], robust-training baseline, *AFR* [22] and standard empirical risk minimization (ERM) baselines.

**Contributions.**

1. We introduce **R4RR**, a VLM-teacher framework that aligns CNN evidence by matching CAMs to CLIP-based semantic affinity maps using KL divergence over spatial distributions.

2. We present a practical two-phase training schedule and show that *Classify-then-Align* is the strongest default, with alternative curricula analyzed in ablation.

3. We demonstrate consistent performance gains on three complementary benchmarks (*DecoyMNIST* [6], *Waterbirds* [26], and *RedMeat* [2]) of spurious correlation.

# 2 Related Work

A prominent line of work improves robustness to shortcut learning by supervising explanations during training. Right for the Right Reasons (RRR) [25] constrains input-gradient explanations by penalizing gradients on features deemed irrelevant. CDEP [24] extends this idea by penalizing feature and interaction attributions derived from contextual decomposition. Concept Distillation [12] similarly uses concept-based explanations for ante-hoc model improvement via concept-sensitive losses.

Beyond gradient and concept penalties, several methods study explanation supervision under noisy labels and incomplete rationales [9], as well as interactive human-in-the-loop correction of model attention [10, 27]. These methods can be effective, but they often require pixel-level masks, bounding boxes, concept libraries, or repeated expert feedback. This creates a supervision bottleneck that can limit scalability across datasets, tasks, and deployment settings.

A broader spurious-correlation robustness literature tackles the problem without directly supervising explanations. Common strategies include optimizing for worst-group performance when group structure is available, reweighting or upsampling hard/minority examples, and learning representations that are less sensitive to shortcut features or domain-specific context. Benchmark settings such as Waterbirds [26] have become central in evaluating these approaches, while methods such as AFR [22] illustrate the appeal of lightweight bias-aware

reweighting when explicit group annotations are unavailable. Relative to this line of work, R4RR uses spatial evidence alignment as a complementary mechanism: rather than only changing which samples are emphasized, it directly shapes where the classifier places class evidence.

In parallel, vision–language learning has shown that language can provide scalable semantic priors. Early zero-shot and attribute-driven approaches condition visual models on textual descriptions to transfer across categories [5, 7, 16, 17, 23]. More recently, CLIP-based weakly supervised localization and segmentation pipelines demonstrate that prompt-conditioned representations can produce useful spatial cues using only image-level supervision [19, 32].

GALS [20] is the closest prior in this language-guided debiasing regime. It grounds task-level language specifications with a pretrained VLM to produce saliency targets, then applies an auxiliary explanation loss based on RRR-style input-gradient constraints. This formulation is elegant and practical, but it still inherits the instability and noise sensitivity of gradient-level supervision in strongly biased settings, especially when classification and attention objectives compete early in optimization.

Other robustness directions are complementary. Attention Branch Network (ABN) [8] adds an attention branch and a perception branch trained jointly, so attention supports both explanation and prediction. Automatic Feature Reweighting (AFR) [22] takes a post-hoc route by retraining the final layer of an ERM model with sample reweighting, improving minority-group performance without explicit group labels. Concept Bottleneck Models (CBMs) [15] and post-hoc CBMs (PCBMs) [31] provide a concept-mediated alternative by routing prediction through interpretable concept spaces.

Our approach (**R4RR**) is most closely related to language-guided attention supervision such as GALS [20], but differs in three key ways. First, GALS enforces guidance through input-gradient regularization, whereas R4RR aligns CAM evidence distributions in latent space, avoiding noisy and expensive pixel-gradient supervision. Second, R4RR introduces a two-phase curriculum that explicitly manages objective competition; this is crucial in spurious-correlation settings where early shortcut acquisition can dominate joint training. Third, R4RR uses the WeCLIP+ [32] pipeline to convert prompts into segmentation-style pseudo masks, whereas GALS uses VLM saliency heatmaps.

# 3 Methodology

## 3.1 Problem Setup and Notation

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ denote a labeled image dataset with $x_i \in \mathbb{R}^{H \times W \times 3}$ and $y_i \in \{1, \ldots, C\}$. We consider settings where images contain *task-irrelevant* regions that are spuriously correlated with the label (e.g., background), which can induce shortcut learning.

We train a CNN classifier $f_\theta$ with parameters $\theta$ that outputs logits $z_\theta(x) \in \mathbb{R}^C$ and predicted probabilities $p_\theta(x) = \mathrm{softmax}(z_\theta(x))$. Our goal is to learn a classifier that achieves strong accuracy while basing its decision primarily on regions that are semantically relevant to the target class, without relying on per-image human masks.

## 3.2 Overview

For each training image $x_i$ with label $y_i$, **R4RR** constructs:

- A **teacher map** $A^{\mathrm{VLM}}(x_i, y_i) \in [0,1]^{H_{\mathrm{img}} \times W_{\mathrm{img}}}$ from a pretrained VLM-based pipeline, intended to highlight task-relevant regions under a natural-language prompt.
- A **student evidence map** $A_\theta^{\mathrm{CNN}}(x_i, y_i) \in [0,1]^{H_{\mathrm{cam}} \times W_{\mathrm{cam}}}$ computed via CAM at the final convolutional resolution, indicating where the CNN allocates evidence for class $y_i$.

We align these maps by treating them as spatial probability distributions on the CAM grid and minimizing a KL divergence objective. This shifts supervision from noisy input gradients (common in RRR-style regularization [25]) to a smoother, feature-based evidence signal.

## 3.3 Teacher: VLM-Derived Semantic Affinity Maps

**Language specification.** We define two term sets: a *foreground* set $\mathcal{C}$ (dataset class names) and a *background* set $\mathcal{B}$ (common context categories). For each term $t \in \mathcal{C} \cup \mathcal{B}$ we instantiate a single prompt template, "`a clean origami of {t}.`" [1] We release dataset-specific foreground/background term lists with our code and supplementary material. A brief overview of task-level prompts is provided in the supplementary material.

[1] WeCLIP+ uses this fixed prompt template and reports strong performance with it; although the phrasing is unconventional compared to "`a photo of {t}`", it serves the same role of wrapping the class term in natural language for CLIP prompting [32].

**Teacher backbone and map generation.** We adopt WeCLIP+ [32] as the teacher, which pairs a frozen CLIP-family image–text encoder with a frozen DINO-family visual encoder [3]. Given an image $x$, the teacher forms an initial class-conditioned localization seed by applying Grad-CAM [28] on the frozen CLIP visual backbone with respect to the foreground/background text prototypes. It then refines this seed using WeCLIP+'s refinement module (RFM+), which fuses CLIP and DINO patch features and leverages CLIP attention to propagate and complete the CAM. Following WeCLIP+, we train only the lightweight refinement heads on the *training split* (with frozen CLIP/DINO backbones), then freeze the entire teacher and generate maps for the train and validation splits (no adaptation on validation/test). All teacher maps are precomputed once and cached; the teacher is not used at inference time for the student.

## 3.4 Student Evidence: Class Activation Maps

We compute student evidence maps using Class Activation Mapping (CAM) [35] at the spatial resolution of the final convolutional block. Unless otherwise noted, we use standard CAM [35] for all ResNet[13] models; for DecoyMNIST [6] we use Grad-CAM [28] (on the second convolutional layer) to match the LeNet-style architecture and prior protocol [24].

Let $F_\theta(x) \in \mathbb{R}^{D\times H_{\text{cam}}\times W_{\text{cam}}}$ denote the last convolutional feature tensor and let $\mathbf{W} \in \mathbb{R}^{C\times D}$ be the linear classifier weights (for $C$ classes). For an image $x$ and class $c$, the (pre-nonlinearity) CAM is

$$\tilde{A}^{\text{CNN}}_\theta(x,c)[u,v] = \sum_{d=1}^{D} \mathbf{W}_{c,d}\, F_\theta(x)[d,u,v], \qquad (u,v) \in \{1,\dots,H_{\text{cam}}\} \times \{1,\dots,W_{\text{cam}}\}. \quad (1)$$

In our implementation, we form the evidence map for the ground-truth class $c = y$.

**Non-negativity and normalization.** We apply a ReLU followed by per-image min–max normalization (matching the code path):

$$R_\theta(x,c)[u,v] = \text{ReLU}\Big(\tilde{A}^{\text{CNN}}_\theta(x,c)[u,v]\Big), \quad (2)$$

$$A^{\text{CNN}}_\theta(x,c)[u,v] = \frac{R_\theta(x,c)[u,v] - \min_{u',v'} R_\theta(x,c)[u',v']}{\max_{u',v'} R_\theta(x,c)[u',v'] - \min_{u',v'} R_\theta(x,c)[u',v'] + \epsilon}. \quad (3)$$

We then convert this map into log-probabilities over spatial locations via a *spatial log-*

*softmax*:

$$\log \hat{A}_\theta^{\mathrm{CNN}}(x,c)[u,v] = A_\theta^{\mathrm{CNN}}(x,c)[u,v] - \log \sum_{u',v'} \exp\big(A_\theta^{\mathrm{CNN}}(x,c)[u',v']\big). \tag{4}$$

**Teacher map preprocessing.** Let $A^{\mathrm{VLM}}(x,y) \in \mathbb{R}_{\geq 0}^{H_{\mathrm{img}} \times W_{\mathrm{img}}}$ be the non-negative teacher map for $x$ and label $y$. To align resolutions, we downsample the teacher map to the CAM resolution using nearest-neighbor interpolation:

$$A_\downarrow^{\mathrm{VLM}}(x,y) = \mathrm{Downsample}_{\mathrm{NN}}\big(A^{\mathrm{VLM}}(x,y)\big) \in \mathbb{R}_{\geq 0}^{H_{\mathrm{cam}} \times W_{\mathrm{cam}}}. \tag{5}$$

We then normalize by its sum to obtain a spatial distribution:

$$\hat{A}^{\mathrm{VLM}}(x,y)[u,v] = \frac{A_\downarrow^{\mathrm{VLM}}(x,y)[u,v]}{\sum_{u',v'} A_\downarrow^{\mathrm{VLM}}(x,y)[u',v'] + \epsilon}. \tag{6}$$

## 3.5 Evidence Alignment Objective

**Classification loss.** We use standard cross-entropy:

$$\mathcal{L}_{\mathrm{CE}}(\theta) = -\frac{1}{B} \sum_{i=1}^{B} \log p_\theta(y_i \mid x_i) \tag{7}$$

**KL evidence alignment loss (teacher → student).** For each sample $(x_i, y_i)$, we align the teacher distribution to the student evidence distribution using KL divergence in the *teacher//student* direction:

$$\mathcal{L}_{\mathrm{align}}(\theta) = \frac{1}{B} \sum_{i=1}^{B} \mathrm{KL}\Big(\hat{A}^{\mathrm{VLM}}(x_i,y_i) \,\Big\|\, \hat{A}_\theta^{\mathrm{CNN}}(x_i,y_i)\Big), \tag{8}$$

where (on the CAM grid), $\mathrm{KL}(P\|Q) = \sum_{h,w} P[h,w] \log \frac{P[h,w]}{Q[h,w]+\epsilon}$.

**Final objective.** The joint training objective is:

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathrm{CE}}(\theta) + \lambda\, \mathcal{L}_{\mathrm{align}}(\theta), \tag{9}$$

where $\lambda > 0$ controls the strength of evidence alignment. We select $\lambda$ via hyperparameter tuning on the validation set (details in the supplementary material).

### 3.6 Two-Phase Training: Classify-then-Align

Directly optimizing Eq. (9) from scratch can be unstable. Classification may quickly latch onto easy shortcuts before alignment can reshape evidence, while strong early alignment can also hinder learning discriminative structure in complex images. We therefore adopt a two-phase *Classify-then-Align* curriculum as our default training protocol.

**Phase 1: Classification warm-up.** We first optimize only the classification objective until the *attention epoch* $E_{\text{cls}}$: $\min_\theta \mathcal{L}_{\text{CE}}(\theta)$. We treat $E_{\text{cls}}$ (the epoch at which we activate evidence alignment) as a tunable hyperparameter selected on the validation set.

**Phase 2: Joint optimization with evidence alignment.** We then optimize the combined objective for $E_{\text{joint}}$ epochs:

$$\min_\theta \ \mathcal{L}_{\text{CE}}(\theta) + \lambda \mathcal{L}_{\text{align}}(\theta). \tag{10}$$

**Model selection.** To ensure the selected checkpoint reflects the effect of evidence alignment, we select the best model using validation performance *only after* the attention epoch (i.e., only once Phase 2 is active), and ignore validation scores obtained during the Phase 1 warm-up. algorithm 1 shows our approach.

## 4 Experiments

### 4.1 Experimental Setup

**Datasets.** We evaluate our approach **(R4RR)** across three widely used benchmarks for spurious correlation: (1) *Waterbirds* [26], where bird type is correlated with background; (2)*DecoyMNIST* [6], a digit-classification task with a strong decoy cue that induces shortcut learning; and (3) *RedMeat* [20], a 5-way classification task constructed from Food-101 [2] that exhibits implicit bias and noisy context. Split details and group/class counts are reported in supplementary material.

**Training setup.** For *Waterbirds* and *RedMeat*, we follow the standard protocol of GALS [20]. We use an ImageNet-pretrained ResNet-50 [13], resize inputs to $224 \times 224$, and apply standard ImageNet normalization. We train with SGD (momentum 0.9, weight decay $10^{-5}$) and batch size 96 for 200 epochs on Waterbirds and 150 epochs on *RedMeat*. Following GALS [20], we use two optimizer parameter groups with separate learning rates: `base_lr` for the pretrained backbone and `classifier_lr` for the final linear head.

**Algorithm 1** R4RR Training with VLM-guided Evidence Alignment (Classify-then-Align)

**Require:** Dataset $\mathcal{D}$, teacher VLM, prompts $\{\mathcal{T}_c\}_{c=1}^{C}$, alignment weight $\lambda$, warm-up epochs $E_{\text{cls}}$, joint epochs $E_{\text{joint}}$

**Precompute teacher maps:** for each $(x_i, y_i) \in \mathcal{D}$ compute $A^{\text{VLM}}(x_i, y_i)$ using VLM and prompts; store maps.
Initialize CNN $f_\theta$.
**Phase 1 (classification):** optimize $\mathcal{L}_{\text{CE}}$ for $E_{\text{cls}}$ epochs.
Reset optimizer and scheduler; set the Phase 2 learning rate to $\eta_{\text{joint}} = m \cdot \eta_{\text{base}}$, where the multiplier $m$ is tuned.
**Phase 2 (joint):** optimize $\mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{align}}$ for $E_{\text{joint}}$ epochs:
**for** epoch $= 1$ to $E_{\text{joint}}$ **do**
  **for** minibatch $\{(x_i, y_i)\}_{i=1}^{B}$ **do**
    Compute logits $z_\theta(x_i)$ and $\mathcal{L}_{\text{CE}}$.
    Compute CAMs $A_\theta^{\text{CNN}}(x_i, y_i)$ and normalize to $\hat{A}_\theta^{\text{CNN}}$.
    Downsample teacher maps $A^{\text{VLM}}(x_i, y_i)$ to resolution $H_{\text{cam}} \times W_{\text{cam}}$ and normalize to $\hat{A}^{\text{VLM}}(x_i, y_i)$.
    Compute $\mathcal{L}_{\text{align}} = \frac{1}{B} \sum_i \text{KL}(\hat{A}^{\text{VLM}}(x_i, y_i) \| \hat{A}_\theta^{\text{CNN}}(x_i, y_i))$.
    Update $\theta$ using $\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{align}}$.
  **end for**
**end for**
**Inference:** use $f_\theta$ only (no teacher required).

For *DecoyMNIST*, we match the setup used by CDEP [24]. We train a lightweight LeNet-style CNN on $28 \times 28$ grayscale inputs normalized to $[-1, 1]$, using Adam with weight decay $10^{-4}$ and a fixed single learning rate of $10^{-3}$. We train for 19 epochs across all methods and select checkpoints by validation accuracy. We choose 19 epochs to approximately match the number of training-image exposures used on *Waterbirds*, i.e., $200 \cdot 4750/50000 \approx 19$. We hold out 10% of the training set for validation and use it for checkpoint selection.

**Additional implementation details.** For Waterbirds and RedMeat, R4RR evidence alignment is activated at the validation-selected attention epoch, and checkpoint selection is performed only after alignment is active. Teacher maps are generated offline once per dataset and cached during training; the full cached map footprint is modest ($\text{Waterbirds}_{95}$: 19 MB, $\text{Waterbirds}_{100}$: 19 MB, RedMeat: 12 MB, DecoyMNIST: 30 MB).

**Baselines.** We compare **R4RR** against methods spanning language-guided attention supervision, architectural attention mechanisms, lightweight robustness reweighting, and VLM baselines:

**1. Standard baselines:** *Vanilla* (standard empirical risk minimization) and *Up-weight* [20], which uses class-balanced cross-entropy with weights inversely proportional to

class frequency in the training set.

**2. VLM-specific baselines:** Zero-shot CLIP ($CLIP_{Zero}$) and a linear probe on CLIP features ($CLIP_{LR}$), i.e., logistic regression trained on frozen **CLIP RN50** [23] image-encoder embeddings following standard practice [23, 20].

**3. GALS** [20]: It uses a pretrained VLM to ground high-level language specifications into per-image saliency maps, and then constrains the classifier via an auxiliary explanation regularization loss that discourages reliance on distracting context.

**4. ABN** [8]: Attention Branch Network (ABN) augments a standard CNN with an attention branch and a perception branch that performs classification using both the feature and attention maps; both branches are trained end-to-end so that attention supports prediction and visual explanation.

**5. AFR** [22]: Automatic Feature Reweighting (AFR) is a post-hoc robustness method that starts from an ERM-trained model and retrains only the final layer using a weighted loss that upweights samples the ERM model predicts poorly, improving group robustness without requiring group labels.

**Evaluation groups.** Across datasets, we report performance over a set of predefined *evaluation groups*. For *Waterbirds*, groups are the four combinations of label and background (landbird/waterbird $\times$ land/water). For *DecoyMNIST*, groups correspond to digit classes; for *RedMeat*, groups correspond to the 5 target classes: baby back ribs, filet mignon, pork chop, prime rib, and steak.

**Metrics.** Following [26], we report performance on the datasets using *group-wise* evaluation. We summarize results with (i) *average group accuracy* ($\mathbf{Per}_{Group}$), which assigns equal weight to each group regardless of frequency, and (ii) *worst-group accuracy*, defined as the minimum accuracy over the four groups ($\mathbf{Worst}_{Group}$). Worst-group accuracy is particularly informative because it reflects performance on the minority and bias-conflicting groups (e.g. *waterbird-on-land* in Waterbirds dataset), which degrade most severely when the model relies on spurious background cues.

**Hyperparameter optimization.** For each dataset and method (unless noted otherwise), we selected hyperparameters using automated validation-based tuning. Specifically, we ran 50 Optuna [1] trials per *(dataset, method)* configuration and chose the single best trial by validation performance. Each trial samples from the search ranges reported in the supplementary material. We applied this protocol separately for Waterbirds, and RedMeat across all baselines and our method. For DecoyMNIST, we transfer hyperparameters optimized on $\text{Waterbirds}_{100}$,

motivated by the fact that both are fully biased training regimes. See supplementary material for further details. We also release the validation-selected hyperparameters for all methods, variants, and ablations for each dataset in our codebase[2] under *configs/*.

## 4.2 Results

**Fair baseline tuning.** To avoid under-tuning any baseline, we optimize each method separately using the same validation-driven protocol. For GALS specifically, we sweep multiple common variants (CLIP RN50 vs. ViT grounding; RRR/ABN/Grad-CAM mechanisms) and report the strongest validation-selected variant per dataset in the main comparisons. The full variant sweep is reported in Appendix F (Table 20).

### 4.2.1 Bias with DecoyMNIST.

*DecoyMNIST* [6] is a variant of MNIST [18], where MNIST digits are augmented by adding class-indicative gray patches to the image boundary. These "decoy" patches create a spurious association between digit class and patch location or intensity.

**Teacher Map Generation.** We use the foreground prompt `"digit"` and choose background prompts to capture the canvas and boundary decoy artifacts (e.g., `"background"`, `"blank"`, `"corner"`, `"patch"`).

Table 1: DecoyMNIST performance evaluation with test accuracy (mean±std over 5 runs) comparing standard ERM trained model (**Vanilla**), class-balanced (**Upweight**), zero-shot CLIP classifier (**CLIP**$_{Zero}$), linear probe on CLIP features (**CLIP**$_{LR}$), **GALS** [20], **ABN** [8], **AFR** [22], and our proposed method (**Ours**). Higher is better.

| | **Vanilla** | **CLIP**$_{Zero}$ | **CLIP**$_{LR}$ | **Upweight** | **ABN** | **GALS** | **AFR** | **Ours** |
|---|---|---|---|---|---|---|---|---|
| **Per**$_{Group}$ | 45.9±2.7 | 55.2±0.0 | 88.3±0.0 | 49.2±3.2 | 80.8±6.0 | 73.2±11.3 | 44.6±3.2 | **96.6±0.1** |
| **Worst**$_{Group}$ | 0.0±0.0 | 0.0±0.0 | 76.2±0.0 | 0.0±0.0 | 25.2±17.0 | 26.9±22.8 | 0.0±0.0 | **95.8±1.3** |

**Evaluation.** Table 1 reports group-wise performance on *DecoyMNIST*, a setting where models can achieve high training accuracy by exploiting a decoy correlation rather than the digit evidence. Standard ERM (**Vanilla**) and class reweighting (**Upweight**) collapse completely on the bias-conflicting groups, indicating near-total reliance on the decoy cue. Zero-shot CLIP (**CLIP**$_{Zero}$) shows the same failure mode on worst-group accuracy despite improved average performance. In contrast, our method achieves the highest per-group and worst-group accuracy, substantially outperforming all baselines we compare against.

[2]We provide code as a part of the supplementary material.

Table 2: Waterbirds performance evaluation with test accuracy (mean±std over 5 runs) comparing standard ERM trained model (**Vanilla**), class-balanced (**Upweight**), zero-shot CLIP classifier (**CLIP**$_{Zero}$), linear probe on CLIP features (**CLIP**$_{LR}$), **GALS** [20], **ABN** [8], **AFR** [22], and our proposed method (**Ours**). Higher is better.

| | | **Vanilla** | **CLIP**$_{Zero}$ | **CLIP**$_{LR}$ | **Upweight** | **ABN** | **GALS** | **AFR** | **Ours** |
|---|---|---|---|---|---|---|---|---|---|
| **Waterbirds$_{95}$** | **Per**$_{Group}$ | 87.1±0.6 | 73.7±0.0 | 80.4±0.0 | 86.7±0.2 | 84.5±1.2 | 89.1±0.4 | **91.9±0.7** | 90.2±0.7 |
| | **Worst**$_{Group}$ | 71.2±0.6 | 44.1±0.0 | 56.2±0.0 | 70.8±2.2 | 62.9±5.0 | 76.0±1.9 | **90.4±1.1** | 87.1±2.7 |
| **Waterbirds$_{100}$** | **Per**$_{Group}$ | 70.2±2.9 | 73.4±0.0 | 63.7±0.0 | 70.9±4.1 | 69.1±2.1 | 79.9±2.1 | 69.6±1.9 | **89.1±0.4** |
| | **Worst**$_{Group}$ | 40.3±5.5 | 43.8±0.0 | 24.6±0.0 | 32.9±10.6 | 34.6±7.1 | 57.2±5.5 | 38.4±4.3 | **84.8±1.0** |

#### 4.2.2 Explicit bias on Waterbirds.

*Waterbirds* [26] is a synthetic bias dataset constructed by segmenting birds from CUB [30] and pasting them onto land or water backgrounds from Places [36]. The labels are bird types (landbird vs. waterbird). During training, most waterbirds appear on water backgrounds and landbirds on land backgrounds, whereas validation and test sets contain an equal number of samples for each class on both backgrounds. We consider two settings: **Waterbirds-95%** (**Waterbirds$_{95}$**), where 5% of training samples counter the bias, and the more challenging **Waterbirds-100%** (**Waterbirds$_{100}$**), where the background and label are perfectly correlated during training.

**Teacher Map Generation.** We use the task-level foreground prompt `"bird"` and a background (distractor) prompt set capturing common *land* and *water* scene context (e.g., shoreline/water terms and terrain/vegetation terms).

**Evaluation.** Table 2 shows that **Vanilla** model performs well on majority groups but suffers substantial degradation on the bias-conflicting minority groups, especially in Waterbirds-100% setting. In contrast, our method delivers the strongest robustness in Waterbirds-100% dataset. On Waterbirds-95%, where some counter-biased samples are available during training, our method remains competitive and substantially improves worst-group performance over **Vanilla** model. While **AFR** attains the highest group accuracy in this easier setting, its performance does not transfer to Waterbirds-100%. In contrast, our method maintains strong worst-group accuracy in both settings, suggesting that teacher-guided evidence alignment provides a more reliable inductive bias when the spurious correlation is strongest.

#### 4.2.3 Red Meat Classification with Noisy Data.

*RedMeat* [20] is a 5-way task constructed from Food-101 [2], with classes *baby back ribs*, *filet mignon*, *pork chop*, *prime rib*, and *steak*. It contains implicit bias and noise due to

Table 3: Redmeat performance evaluation with test accuracy (mean±std over 5 runs) comparing standard ERM trained model (**Vanilla**), class-balanced (**Upweight**), zero-shot CLIP classifier (**CLIP**$_{Zero}$), linear probe on CLIP features (**CLIP**$_{LR}$), **GALS** [20], **ABN** [8], **AFR** [22], and our proposed method (**Ours**). Higher is better.

| | **Vanilla** | **CLIP**$_{Zero}$ | **CLIP**$_{LR}$ | **Upweight** | **ABN** | **GALS** | **AFR** | **Ours** |
|---|---|---|---|---|---|---|---|---|
| **Per**$_{Group}$ | 67.3±0.3 | 61.5±0.0 | **76.6±0.0** | 68.3±1.2 | 65.6±0.3 | 71.9±0.3 | 60.7±1.9 | <u>73.2±1.0</u> |
| **Worst**$_{Group}$ | 48.0±4.5 | 5.6±0.0 | 54.0±0.0 | 47.9±4.8 | 55.3±4.1 | 57.0±1.5 | 58.2±2.4 | **60.9±3.4** |

intentionally uncleaned training images (e.g., label noise, bright colors, and co-occurring context such as sauces). In contrast, the test split is intentionally curated to be cleaner, so models that rely on these training-only artifacts tend to generalize poorly when those cues are absent at test time.

**Teacher Map Generation.** We use the foreground prompt `"meat"` and background prompts for common food-scene context (e.g., plates, utensils, sauces, sides), encouraging localization on the meat rather than co-occurring context.

**Evaluation.** Table 3 summarizes performance on *RedMeat*. **Vanilla** model attains reasonable per-group accuracy but exhibits a large drop in the worst-group performance. Our method achieves the best worst-group accuracy while also improving per-group accuracy over **Vanilla** model and class-balancing technique (**Upweight**). A linear probe on frozen CLIP features (**CLIP**$_{LR}$) improves per-group performance substantially obtaining the best performance, but still lags behind our method on the worst-group, suggesting that directly *training* the classifier to allocate evidence to relevant regions provides benefits beyond strong representations alone.

#### 4.2.4 Qualitative Evidence Alignment

Figures 2, 3, and 4 show qualitative saliency comparisons on all three datasets. All saliency maps in this subsection are generated using RISE [21].

### 4.3 Attention evaluation with the Pointing Game

Beyond classification performance, we assess whether our training procedure improves *where* the model attends, i.e., whether its explanations concentrate on task-relevant evidence. We use the *Pointing Game* [34], a standard protocol for evaluating spatial explanations. For each input image $x_i$, the Pointing Game requires (i) an explanation map $a_i \in \mathbb{R}^{H \times W}$ produced by

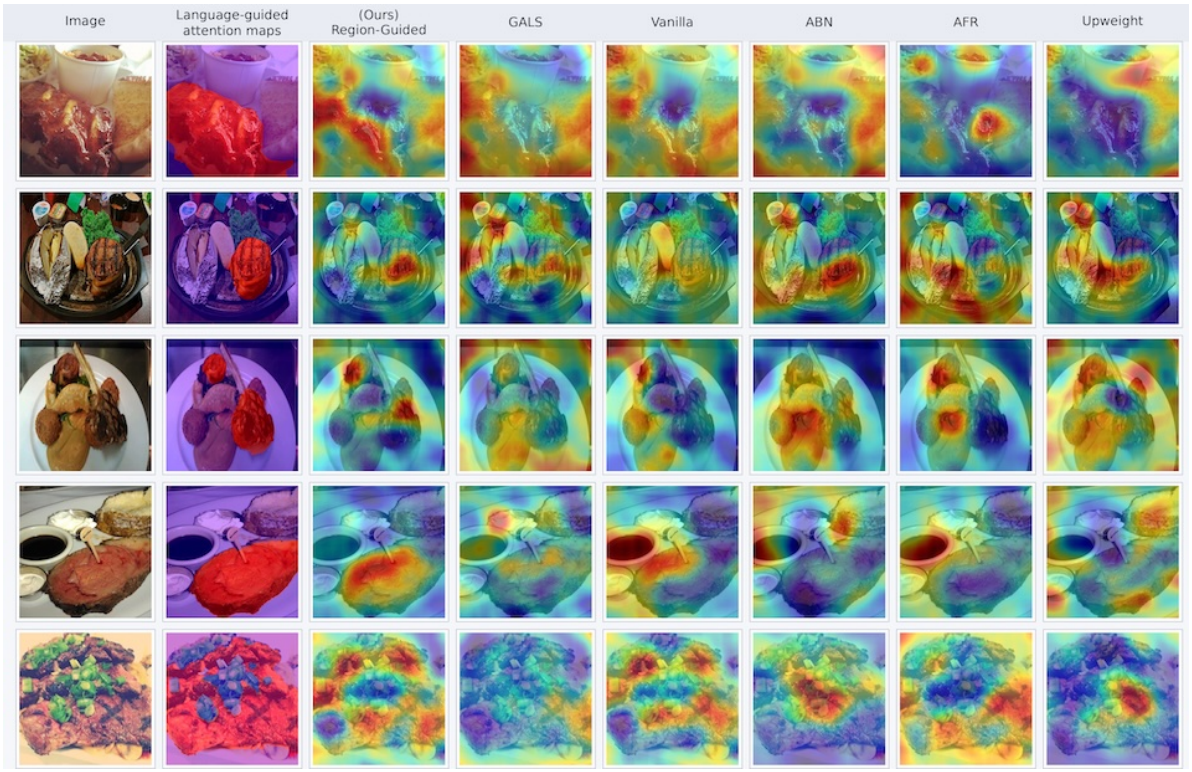


Figure 2: **Right for the wrong regions vs. right for the right regions on RedMeat.** Comparing saliency maps between our method, *GALS* [20], standard *vanilla* model, *ABN* [8], *AFR* [22] and class balanced model (*Upweight*). Shortcut-prone models often attend to spurious context (e.g., background).

the model, and (ii) a binary mask $Z_i \in \{0,1\}^{H\times W}$ indicating task-relevant pixels. A sample is counted as a *hit* if the location of the maximum explanation value lies inside the mask. The Pointing Game score is the average hit rate over the evaluation set. Intuitively, a higher score indicates that the model's peak evidence aligns with the relevant region, suggesting that the model is "right for the right regions" rather than relying on spurious context. We use the bird segmentation masks in Waterbirds-95% and Waterbirds-100% to define $Z_i$.

Table 4 shows that our method produces the most faithful spatial evidence among all compared methods, achieving the highest Pointing Game hit rate on both Waterbirds settings. The improvement is especially pronounced on the fully biased Waterbirds-100% split, where all other models have reduced performance.

Table 4: Pointing Game evaluation on Waterbirds with hit rate (%) measuring whether the peak CAM activation falls inside the ground-truth bird segmentation mask, comparing standard ERM trained model (**Vanilla**), class-balanced (**Upweight**), zero-shot CLIP classifier (**CLIP**$_{Zero}$), linear probe on CLIP features (**CLIP**$_{LR}$), **GALS** [20], **ABN** [8], **AFR** [22], and our proposed method (**Ours**). Higher indicates better localization of task-relevant evidence.

| | **Vanilla** | **CLIP**$_{Zero}$ | **CLIP**$_{LR}$ | **Upweight** | **ABN** | **GALS** | **AFR** | **Ours** |
|---|---|---|---|---|---|---|---|---|
| Waterbirds$_{\mathbf{95}}$ | 59.9 | 32.4 | 53.5 | 67.4 | 7.8 | 69.4 | 63.9 | **76.6** |
| Waterbirds$_{\mathbf{100}}$ | 46.5 | 32.0 | 36.4 | 59.3 | 6.5 | 59.3 | 61.6 | **74.7** |

# 5 Ablation studies

## 5.1 Study on training curriculum

R4RR optimizes a classification loss $\mathcal{L}_{\text{CE}}$ and an evidence-alignment loss $\mathcal{L}_{\text{align}}$. Our default curriculum is **Classify→Align**: we warm-start with $\mathcal{L}_{\text{CE}}$ for $E_{\text{cls}}$ epochs, then optimize $\mathcal{L}_{\text{CE}} + \lambda\mathcal{L}_{\text{align}}$. We compare against two alternatives on Waterbirds: **Align→Classify** and **Joint** (optimize the joint loss from scratch).

Table 5: Ablation on training curriculum on Waterbirds, comparing three schedules: **Align→Classify**, **Joint**, and **Classify→Align (ours)**. Higher is better.

| | | **Align→Classify** | **Joint** | **Classify→Align (ours)** |
|---|---|---|---|---|
| **Waterbirds$_{\mathbf{95}}$** | **Per**$_{Group}$ | 86.3±0.6 | 85.5±0.9 | **90.2**±0.7 |
| | **Worst**$_{Group}$ | 67.6±2.2 | 65.9±3.7 | **87.1**±2.7 |
| **Waterbirds$_{\mathbf{100}}$** | **Per**$_{Group}$ | 74.5±2.5 | 81.5±3.9 | **89.1**±0.4 |
| | **Worst**$_{Group}$ | 44.6±8.4 | 63.4±7.2 | **84.8**±1.0 |

Table 5 shows that the schedule strongly impacts robustness, especially in Waterbirds$_{100}$. **Joint** outperforms **Align→Classify**, indicating that alignment-only warm-up can constrain learning. Our **Classify→Align** is best across both settings consistent with the view that a brief classification warm-start provides a stable representation that alignment can then refine away from background shortcuts.

## 5.2 Study on teacher selection

Our default teacher follows WeCLIP+ [32], which combines CLIP and DINO. Here, we test the sensitivity of **R4RR** to the teacher by swapping (i) the VLM backbone (CLIP → OpenCLIP [14] or SigLIP2 [29]) and (ii) the refinement backbone (DINO → XCiT-DINO [4, 3]), while keeping the student, prompts, loss, and curriculum fixed.

Table 6 shows that **R4RR** is largely insensitive to the teacher choice: all variants maintain

Table 6: Teacher ablation on Waterbirds: VLM & refinement backbones, keeping the student, loss, and curriculum fixed. Results are accuracy (mean±std). Higher is better.

| Teacher variant | Teacher components | | $\text{Waterbirds}_{95}$ | | $\text{Waterbirds}_{100}$ | |
|---|---|---|---|---|---|---|
| | VLM | Refine | $\text{Per}_{Group}$ | $\text{Worst}_{Group}$ | $\text{Per}_{Group}$ | $\text{Worst}_{Group}$ |
| **Baseline (ours)** | CLIP | DINO | $90.2_{\pm 0.7}$ | $87.1_{\pm 2.7}$ | $89.1_{\pm 0.4}$ | $84.8_{\pm 1.0}$ |
| OpenCLIP teacher | OpenCLIP | DINO | $90.5_{\pm 0.3}$ | $88.1_{\pm 0.9}$ | $89.1_{\pm 0.5}$ | $85.6_{\pm 1.8}$ |
| SigLIP2 teacher | SigLIP2 | DINO | $90.3_{\pm 0.3}$ | $87.4_{\pm 0.9}$ | $88.0_{\pm 0.8}$ | $81.6_{\pm 0.8}$ |
| XCiT-DINO refine | CLIP | XCiT-DINO | $90.4_{\pm 0.3}$ | $88.0_{\pm 1.2}$ | $89.1_{\pm 0.1}$ | $84.0_{\pm 1.3}$ |

Table 7: **GALS vs. R4RR component ablation.** Test accuracy (mean±std over 5 runs). Each configuration receives 50 Optuna [1] trials under its own map source and objective/schedule.

| | $\text{Waterbirds}_{95}$ | | $\text{Waterbirds}_{100}$ | | RedMeat | |
|---|---|---|---|---|---|---|
| Scheme | $\text{Per}_{Group}$ | $\text{Worst}_{Group}$ | $\text{Per}_{Group}$ | $\text{Worst}_{Group}$ | $\text{Per}_{Group}$ | $\text{Worst}_{Group}$ |
| (A) | $87.0 \pm 0.4$ | $70.4 \pm 2.6$ | $73.2 \pm 2.2$ | $40.9 \pm 9.4$ | $71.8 \pm 0.7$ | $57.4 \pm 2.2$ |
| (B) | $89.4 \pm 0.3$ | $83.8 \pm 1.5$ | $87.7 \pm 0.3$ | $76.2 \pm 1.7$ | $68.9 \pm 1.5$ | $52.3 \pm 6.2$ |
| (C) | $86.1 \pm 1.5$ | $71.6 \pm 4.7$ | $85.4 \pm 0.1$ | $72.6 \pm 2.0$ | $67.4 \pm 1.1$ | $51.0 \pm 2.4$ |
| (D) | $90.2 \pm 0.7$ | $87.1 \pm 2.7$ | $89.1 \pm 0.5$ | $85.6 \pm 1.8$ | $73.2 \pm 1.0$ | $60.9 \pm 3.4$ |

**Schemes.** (A) *GALS objective + our teacher maps* (RRR-style regularization with WeCLIP+ maps). (B) *R4RR objective/schedule + GALS RN50 maps.* (C) *R4RR objective/schedule + GALS ViT maps.* (D) *R4RR (ours)*: R4RR objective/schedule + our teacher maps.

strong $\text{Per}_{Group}$ accuracy in both bias settings. OpenCLIP is consistently best (or tied) on $\text{Worst}_{Group}$, while SigLIP2 drops on $\text{Waterbirds}_{100}$, suggesting that teacher calibration matters most when training is fully biased. This trend largely carries over to other datasets, discussed in the supplementary material.

## 5.3 Ablation: teacher maps vs. objective/schedule

To disentangle why **R4RR** improves over GALS [20], we factor the gain into (i) the *training mechanism* (our KL evidence-alignment with a two-phase warm-start) and (ii) the *teacher map construction* (WeCLIP+ [32]).

Table 7 shows that map quality alone is not sufficient: **GALS + our maps (A)** remains brittle under full bias ($\text{Waterbirds}_{100}$ worst-group 40.9). In contrast, swapping *only* the training mechanism (**R4RR objective + GALS maps (B–C)**) recovers most of the robustness on Waterbirds, indicating that aligning *latent* evidence under a warm-started two-phase schedule is the primary driver. Finally, **R4RR** with WeCLIP+ maps performs best across datasets, suggesting the two components are complementary: our schedule provides

Table 8: **Teacher-map corruption stress test.** Test accuracy (mean±std over 5 seeds) for **R4RR** under fixed corruption of 15% of training teacher maps. *Clean* denotes the standard unmodified teacher maps used in the main paper. *Blur 15%* applies strong Gaussian blur to a fixed subset of teacher maps, while *Invert 15%* replaces a fixed subset with inverted maps. Higher is better.

| **Dataset** | **Teacher-map condition** | $\mathbf{Per}_{Group}$ | $\mathbf{Worst}_{Group}$ |
|---|---|---|---|
| DecoyMNIST | Clean | 96.6±0.1 | 95.8±1.3 |
| | Blur 15% | 96.7±0.2 | 95.5±0.7 |
| | Invert 15% | 96.8±0.2 | 95.6±0.8 |
| RedMeat | Clean | 73.2±1.0 | 60.9±3.4 |
| | Blur 15% | 72.6±1.5 | 58.7±4.6 |
| | Invert 15% | 72.8±1.1 | 59.4±5.2 |
| $\text{Waterbirds}_{100}$ | Clean | 89.1±0.4 | 84.8±1.0 |
| | Blur 15% | 88.8±0.5 | 85.0±1.4 |
| | Invert 15% | 88.7±0.4 | 84.0±0.8 |
| $\text{Waterbirds}_{95}$ | Clean | 90.2±0.7 | 87.1±2.7 |
| | Blur 15% | 90.2±0.3 | 88.0±0.9 |
| | Invert 15% | 90.1±0.3 | 87.0±0.8 |

the main robustness gain, while segmentation-style pseudo-masks further strengthen the alignment target, especially in the hardest (fully biased) and noisy settings.

## 5.4 Teacher-map corruption stress test

Because **R4RR** relies on precomputed VLM-derived teacher maps, an important question is how sensitive the method is to imperfections in those maps. We evaluate this with a *teacher-map corruption stress test*: for each dataset, we corrupt a fixed 15% subset of training teacher maps once before training and keep them fixed throughout training.

**Setup.** Starting from the validation-selected R4RR configuration, we test two corruption types: **Blur 15%** (aggressive Gaussian blur with renormalization) and **Invert 15%** (map inversion $1 - M$, then renormalization). All other training settings are unchanged (student, optimizer, curriculum, and model selection protocol). We report mean±std over the same five seeds used in the main results.

Table 8 shows that **R4RR** remains strong under moderate teacher corruption. Across all

four dataset settings, corrupting 15% of training maps causes only limited changes relative to the clean teacher baseline, with no collapse on Waterbirds or DecoyMNIST. The largest drop appears on RedMeat, consistent with its noisier and less structured setting. Overall, these results indicate that R4RR is robust to moderate imprecision and partial corruption in teacher supervision.

# 6 Limitations

R4RR relies on VLM-derived teacher maps that are generated outside the main student-training loop. In our current implementation, these maps are precomputed and cached for efficiency, which works well for the datasets studied here and keeps training practical. However, this design can become less convenient as dataset size, image resolution, or teacher-map complexity grows. In principle, teacher maps could also be generated on-the-fly rather than stored ahead of time. This would remove the need to cache the full set of maps, but it introduces a different cost: if the maps do not fit in memory or cache and must be recomputed repeatedly during training, wall-clock time can increase substantially. Thus, the current approach involves a tradeoff between storage efficiency and training speed.

R4RR also depends on the quality and coverage of the VLM teacher. If the teacher systematically localizes the wrong regions, misses task-relevant evidence, or reflects biases from its own pretraining data, the student may inherit those errors through the alignment objective. More generally, the present experiments evaluate R4RR on three controlled spurious-correlation benchmarks. While these datasets provide complementary tests of shortcut reliance, they do not capture the full range of real-world distribution shifts, where biases may be weaker, overlapping, or less cleanly defined.

# 7 Conclusion and Future Work

In this work, we presented Right for the Right Regions (R4RR), a scalable framework that reduces shortcut reliance by aligning a student CNN's latent spatial evidence with task-relevant regions provided by a frozen vision–language teacher. By shifting supervision from noisy input-gradient constraints to distributional alignment of spatial evidence, and by introducing a two-phase Classify-then-Align curriculum, R4RR consistently improved robustness and generalization across three complementary spurious-correlation benchmarks: Waterbirds, DecoyMNIST, and RedMeat. These results suggest that directly shaping where a model places class evidence can be an effective mechanism for reducing reliance on spurious cues.

An important direction for future work, and one that we are actively pursuing, is to extend R4RR beyond CNN-based students to Vision Transformers and other modern vision architectures. In the current formulation, student evidence is represented using CAM-based maps, which are well suited to convolutional networks but do not directly carry over to transformer models. Applying the same core idea in ViTs therefore requires a different notion of spatial class evidence, such as token-level attribution methods, transformer-specific saliency mechanisms, or learned projections from token representations to spatial evidence distributions. More broadly, this ongoing work will help evaluate whether the central principle of latent evidence alignment continues to hold across a wider range of model families, rather than only CNN-based students.

A second direction is to test R4RR in broader and more realistic spurious-correlation settings. Although the benchmarks used here are diverse and challenging, they remain relatively controlled compared to many real-world deployments, where shortcut cues may be weaker, multi-factorial, or less cleanly defined. Evaluating R4RR on larger-scale datasets, more naturalistic distribution shifts, and settings with more ambiguous or mixed sources of bias would help clarify how broadly the method generalizes in practice. Together, these directions would help determine whether R4RR is best understood as a CNN-specific training recipe or as a more general framework for evidence-guided robust learning.

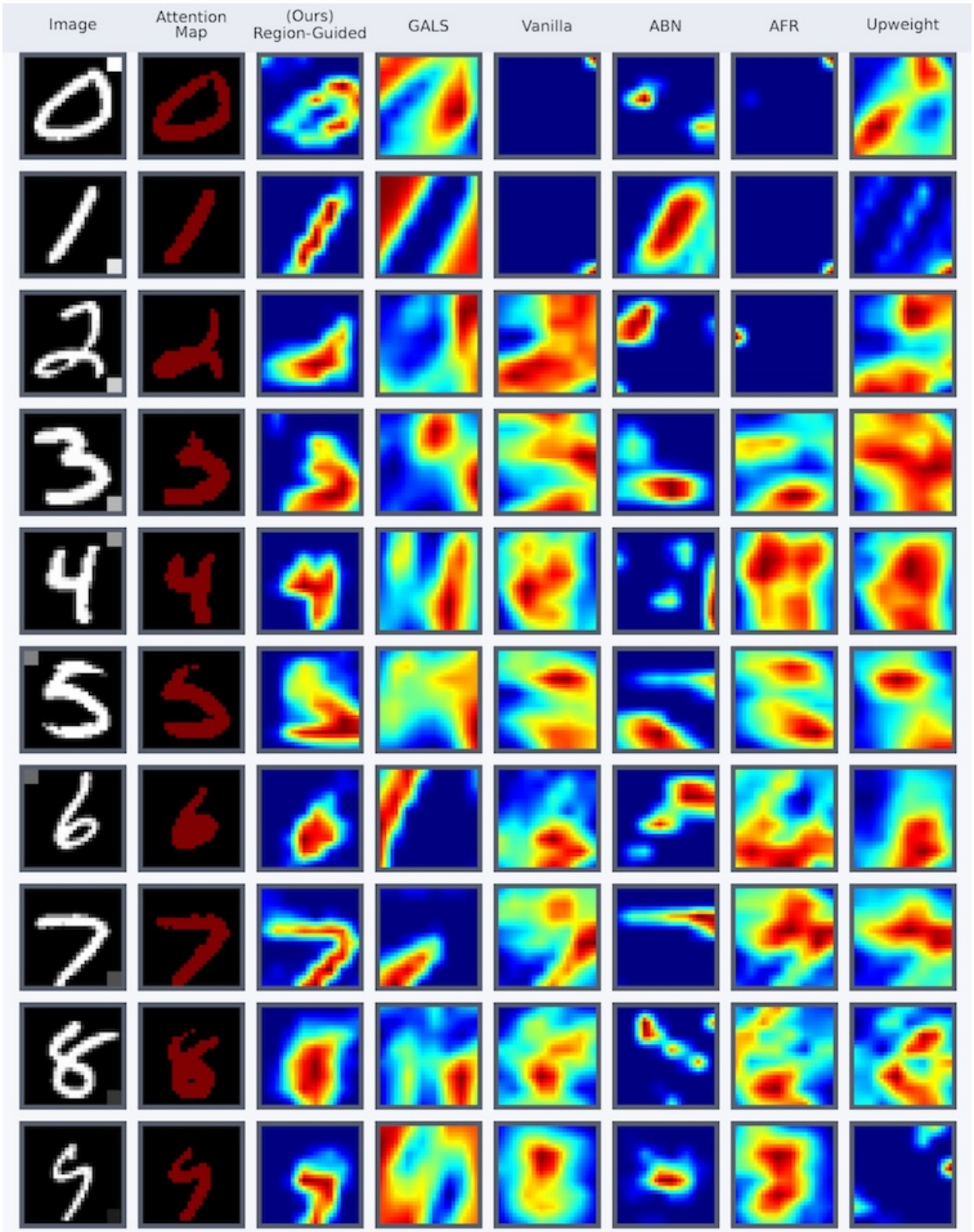


Figure 3: **Right for the wrong regions vs. right for the right regions on DecoyMNIST.** Comparing saliency maps between our method, *GALS* [20], standard *vanilla* model, *ABN* [8], *AFR* [22] and class balanced model (*Upweight*). Shortcut-prone models often attend to spurious context (e.g., background).

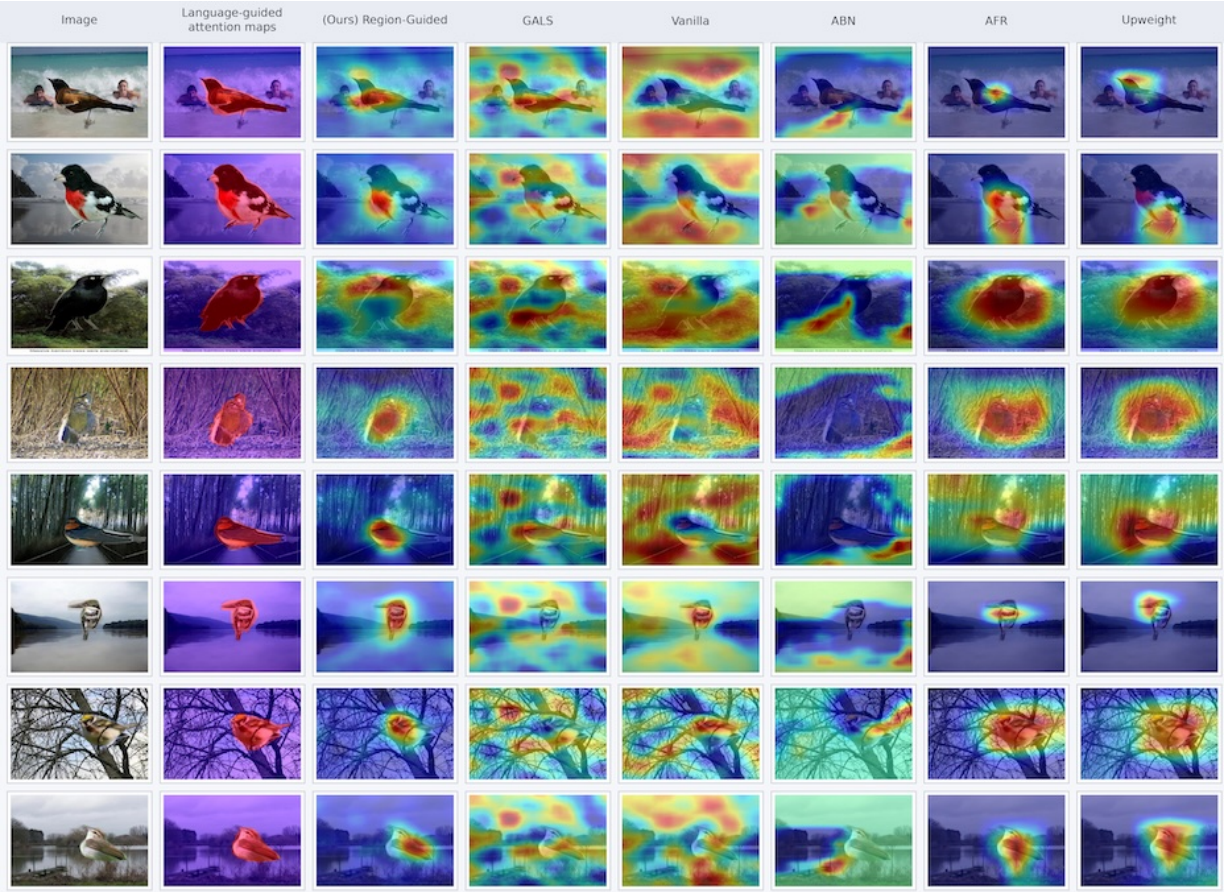


Figure 4: **Right for the wrong regions vs. right for the right regions on Waterbirds.** Comparing saliency maps between our method, *GALS* [20], standard *vanilla* model, *ABN* [8], *AFR* [22] and class balanced model (*Upweight*). Shortcut-prone models often attend to spurious context (e.g., background).

# Appendix

**Table of Contents**


- A Prompts ..... 22
- B Implementation Details ..... 23
- B.2 Teacher map storage ..... 24
- B.3 Hyperparameter Optimization ..... 24
- B.4 R4RR Optimized Hyperparameters ..... 27
- C Local sensitivity to R4RR hyperparameters ..... 28
- D Teacher ablation on additional datasets ..... 29
- E Dataset splits ..... 30
- F Additional GALS Optimization ..... 31


## A Prompts

Table 9: Task-level prompts and background (distractor) prompt lists used per dataset. Background prompt lists are provided in our repository under `WeCLIPPlus/clip/clip_texts/`.

| Dataset | Task-level prompt | Background prompt file |
|---|---|---|
| Waterbirds | `"a clean origami of bird"` | `clip_text_waterbird.py` |
| RedMeat | `"a clean origami of meat"` | `clip_text_redmeat.py` |
| DecoyMNIST | `"a clean origami of digit"` | `clip_text_mnist.py` |

**Background prompts.** We additionally use a set of *background* (distractor) prompts to represent common contextual regions that frequently co-occur with the foreground object and may serve as spurious cues.

**Selecting background prompts.** Background prompt selection is lightweight and dataset-driven. For datasets that are predominantly outdoor (e.g., Waterbirds), we include distractors that commonly appear in outdoor scenes (e.g., terrain and natural context). For primarily indoor datasets, we instead include indoor-relevant distractors (e.g., furniture and man-made interiors). We also reuse the general-purpose background terms provided in our repository as a strong default; in practice these general distractors are effective and reduce the need to hand-engineer dataset-specific lists.

**Why use the prompt "`a clean origami of {t}.`"?** We adopt the fixed prompt wrapper "`a clean origami of {t}.`" because our teacher-map generator follows the same CLIP-based weakly supervised segmentation pipeline used by WeCLIP+ [32]. In this line of work, prompt wording is part of the pseudo-label generator itself, and prior CLIP-WSSS work reports that prompt choice can materially affect the quality of CLIP Grad-CAM pseudo labels. In particular, the original CLIP-based segmentation method underlying this family found that prompt modifiers such as `clean` could improve segmentation quality and selected "`a clean origami of {}.`" as its fixed template [19]. We therefore keep that template unchanged when constructing teacher maps, so that our study isolates the effect of **R4RR**'s evidence-alignment objective and training curriculum rather than introducing an additional prompt-engineering variable.

# B Implementation Details

## B.1 Model Details by Dataset

**Waterbirds.**

- **Student:** ResNet-50[13] (ImageNet pretrained), inputs resized to $224 \times 224$ with ImageNet normalization.
- **Training:** SGD (momentum 0.9, weight decay $10^{-5}$), batch size 96, 200 epochs.

R4RR uses CAM from the final convolutional block; teacher maps are resized to the CAM grid. Alignment is enabled only after the warm-start phase (attention epoch). Model selection uses validation performance computed after alignment is introduced.

**RedMeat.**

- **Student:** ResNet-50[13] (ImageNet pretrained), $224 \times 224$ with ImageNet normalization.

- **Training:** SGD (momentum 0.9, weight decay $10^{-5}$), batch size 96, 150 epochs.

As in Waterbirds, R4RR uses standard CAM from the final convolutional block, with teacher maps resized to the CAM grid. R4RR follows the same Classify→Joint schedule and validation protocol as Waterbirds.

**DecoyMNIST.**

- **Student:** LeNet-style CNN on $28 \times 28$ grayscale inputs normalized to $[-1, 1]$.
- **Training:** Adam (weight decay $10^{-4}$), learning rate $10^{-3}$, 19 epochs (for all methods), model selection by validation accuracy.

Evidence maps use Grad-CAM on the second convolutional layer. The 19-epoch budget matches Waterbirds training exposure: $200 \cdot 4750/50000 \approx 19$. R4RR again uses Classify→Joint, with selection performed after alignment is enabled.

## B.2 Teacher map storage

R4RR uses fixed VLM-derived teacher maps that are generated offline before student training. For hyperparameter optimization, these maps are cached once per dataset and then reused across the full set of Optuna sweeps, so the teacher preprocessing cost is not incurred separately for each hyperparameter trial. In practice, this one-time caching strategy is the natural deployment setting: once the teacher maps are generated, they can be reused throughout tuning and training. For the final five-seed evaluations, however, we regenerate teacher maps separately for each seed and pair them with the corresponding seeded run. This is not required for efficiency, but is a stricter fairness check: it verifies that the reported gains are not tied to a single fixed teacher-map randomization and continue to hold under different seeded instantiations of the teacher preprocessing pipeline.

The resulting cached teacher maps have a modest storage footprint. Table 10 reports the total on-disk size for the full set of generated teacher maps for each dataset, not for a single image. Across all datasets, the cached supervision remains lightweight to store and reuse.

## B.3 Hyperparameter Optimization

**Protocol (overview).** Unless stated otherwise, we tune hyperparameters *separately* for each dataset ($\text{Waterbirds}_{95}$, $\text{Waterbirds}_{100}$, RedMeat) and *separately* for each method and configuration (e.g., each GALS variant is swept independently). For most methods, we run 50 Optuna trials [1] per *(dataset, method)* setting and choose the best trial using validation

Table 10: **Cached teacher-map storage.** Total on-disk size for the full set of generated teacher maps for each dataset.

| Dataset | Total cached size |
|---|---|
| $\text{Waterbirds}_{95}$ | 19 MB |
| $\text{Waterbirds}_{100}$ | 19 MB |
| RedMeat | 12 MB |
| DecoyMNIST | 30 MB |

performance. On Waterbirds, we select by group-aware validation accuracy ($\text{Per}_{Group}$); on RedMeat, we select by standard validation accuracy for the 5-way task. Reported test numbers are mean±std over five seeds (0–4). Experiments were run on a university HPC cluster with NVIDIA A100 PCIe 40 GB GPUs and Intel Xeon Gold 6150 CPUs (2.70 GHz).

For GALS [20], we tune both the default configuration and several design variants under the same Optuna budget, including (i) CLIP ViT vs. CLIP RN50 as the language-attention source and (ii) different classifier-attention mechanisms (RRR, Grad-CAM, and ABN). Unless otherwise stated, we report the strongest tuned GALS variant per dataset under this common budget.

See Table 11 and 12.

Table 11: Hyperparameter tuning summary. Search spaces S1–S9 are defined in Table 12. Unless noted, `weight_decay` $= 10^{-5}$ is fixed by the method configuration. Most deep methods use 50 Optuna trials [1] per dataset; CLIP-ZS is not tuned.

| Family | Variant | Tuned | Space |
|---|---|---|---|
| GALS (RRR-style) | RN50 | `base_lr`, `classifier_lr`, `grad_weight`, `grad_criterion` | S1 |
| | ViT | `base_lr`, `classifier_lr`, `grad_weight`, `grad_criterion` | S1 |
| | OurMasks | `base_lr`, `classifier_lr`, `grad_weight`, `grad_criterion` | S1 |
| GALS+GradCAM | ViT | `base_lr`, `classifier_lr`, `cam_weight` | S2 |
| GALS+ABN | ViT | `base_lr`, `classifier_lr`, `abn_att_weight` | S3 |
| Upweight | — | `base_lr`, `classifier_lr` | S4 |
| ABN baseline | — | `base_lr`, `classifier_lr`, `abn_cls_weight` | S5 |
| Vanilla CNN | — | `base_lr`, `classifier_lr`, `momentum` | S6 |
| Ours (R4RR) | OurMasks | `attention_epoch`, `kl_lambda`, `base_lr`, `classifier_lr`, `lr2_mult` | S7 |
| | RN50 | `attention_epoch`, `kl_lambda`, `base_lr`, `classifier_lr`, `lr2_mult` | S7 |
| | ViT | `attention_epoch`, `kl_lambda`, `base_lr`, `classifier_lr`, `lr2_mult` | S7 |
| CLIP+LR | — | `C` | S8 |
| CLIP-ZS | — | (none; fixed evaluation) | — |
| AFR (two-stage grid; not Optuna) | — | `gamma`, `reg_coeff` | S9 |

Table 12: Search space definitions referenced in Table 11. Ranges apply when running Optuna sweeps, unless the method is explicitly noted as a grid (AFR). For `attention_epoch`, we sweep over valid epoch indices: 0..199 for Waterbirds (200 epochs) and 0..149 for RedMeat (150 epochs).

| ID | Definition |
|---|---|
| S1 | `base_lr` $\in [10^{-5}, 5\cdot10^{-2}]$ (log), `classifier_lr` $\in [10^{-5}, 5\cdot10^{-2}]$ (log), `grad_weight` $\in [10^{3}, 10^{5}]$ (log), `grad_criterion` $\in$ {`L1`, `L2`} |
| S2 | `base_lr` $\in [5\cdot10^{-4}, 5\cdot10^{-2}]$ (cfg-centered), `classifier_lr` $\in [10^{-5}, 10^{-3}]$ (cfg-centered), `cam_weight` $\in [10^{-2}, 10^{2}]$ (log) |
| S3 | `base_lr` $\in [10^{-3}, 10^{-1}]$, `classifier_lr` $\in [10^{-4}, 10^{-2}]$, `abn_att_weight` $\in [10^{-2}, 10^{2}]$ (log) |
| S4 | `base_lr` $\in [5\cdot10^{-5}, 10^{-1}]$, `classifier_lr` $\in [5\cdot10^{-5}, 10^{-1}]$ |
| S5 | `base_lr` $\in [5\cdot10^{-5}, 10^{-1}]$, `classifier_lr` $\in [5\cdot10^{-5}, 10^{-1}]$, `abn_cls_weight` $\in [10^{-2}, 10^{2}]$ (log) |
| S6 | `base_lr` $\in [10^{-5}, 5\cdot10^{-2}]$ (log), `classifier_lr` $\in [10^{-5}, 5\cdot10^{-2}]$ (log), `momentum` $\in [0.85, 0.95]$ |
| S7 | `attention_epoch` $\in \{0, \ldots, E-1\}$ ($E$=200 Waterbirds, $E$=150 RedMeat), `kl_lambda` $\in [1, 500]$ (log), `base_lr` $\in [10^{-5}, 5\cdot10^{-2}]$ (log), `classifier_lr` $\in [10^{-5}, 5\cdot10^{-2}]$ (log), `lr2_mult` $\in [10^{-1}, 3]$ (log) |
| S8 | `C` $\in [10^{-2}, 10^{2}]$ (log) |
| S9 | (AFR grid; not Optuna) full-paper grid: `gamma` $\in \{4.0, 4.5, \ldots, 20.0\}$ (33 values), `reg_coeff` $\in \{0, 0.1, 0.2, 0.3, 0.4\}$ (5 values), for $33 \times 5 = 165$ stage-2 configurations per seed. Reduced grids are also supported for debugging. |

**AFR tuning protocol.** AFR [22] follows a two-stage procedure that differs from the Optuna sweeps above. For each dataset and seed, we (i) train a standard ERM backbone once, then (ii) freeze the backbone and retrain only the final layer while exhaustively sweeping a fixed grid over (`gamma`, `reg_coeff`). A "trial" corresponds to one (`gamma`, `reg_coeff`) pair for one seed (stage-2 only). We use the full grid with `gamma` $\in \{4.0, 4.5, \ldots, 20.0\}$ and `reg_coeff` $\in \{0, 0.1, 0.2, 0.3, 0.4\}$, totaling 165 stage-2 configurations per seed. Within each seed, we select the grid point that maximizes validation worst-group accuracy (`best_val_wga`) and report the corresponding test performance at that validation-selected point (`best_test_at_val`); we then aggregate mean±std over seeds 0–4.

**DecoyMNIST tuning protocol.** We do not tune hyperparameters directly on DecoyMNIST because, in our setting, a validation split drawn from the DecoyMNIST training data is not a reliable proxy for the test-time objective. Since the training set remains fully biased, model selection within that same distribution can favor configurations that fit the decoy shortcut rather than those that genuinely improve robustness under shift. Instead, we treat DecoyMNIST as an out-of-distribution stress test and transfer hyperparameters from $\text{Waterbirds}_{100}$, motivated by the fact that both correspond to fully biased training regimes. This choice is intended to be conservative rather than opportunistic: we do not perform dataset-specific hyperparameter search for our method on DecoyMNIST, and the resulting performance should therefore be interpreted as transfer from one fully biased benchmark to another rather than as the outcome of DecoyMNIST-specific tuning. To match compute across datasets, we also align the approximate number of training-image exposures rather than epochs. Because DecoyMNIST uses a single global learning rate, we fix it to $10^{-3}$ and

set `attention_epoch` by exposure matching. For DecoyMNIST, the reported GALS result uses the same transferred GALS variant selected on $\text{Waterbirds}_{100}$, namely the CLIP ViT + RRR-style configuration.

## B.4 R4RR Optimized Hyperparameters

**Reproducibility (full tuned configs).** For full reproducibility, we release the validation-selected hyperparameters for **all** methods, variants, and ablations (for each dataset) in our codebase under `configs/`. This directory contains dataset- and method-specific YAML configuration files corresponding to the runs reported in the main paper and supplementary material.

**Best R4RR hyperparameters.** Table 13 reports the validation-selected R4RR hyperparameters used for our main results. For Waterbirds and RedMeat, we report the single best Optuna trial per dataset. For DecoyMNIST, we do not tune; we reuse the $\text{Waterbirds}_{100}$ configuration and set `attention_epoch` by matching training-image exposures (19 total epochs $\Rightarrow$ `attention_epoch`=7).

`lr2_mult`. We use a two-phase schedule where Phase 2 (joint optimization with evidence alignment) resets the optimizer. `lr2_mult` is a multiplicative factor applied to the learning rates at the start of Phase 2, i.e., $\texttt{base\_lr}^{(2)} = \texttt{lr2\_mult} \cdot \texttt{base\_lr}$ and $\texttt{classifier\_lr}^{(2)} = \texttt{lr2\_mult} \cdot \texttt{classifier\_lr}$, allowing Phase 2 to run with a smaller or larger step size than Phase 1.

Table 13: **R4RR best hyperparameters.** Best validation-selected configuration per dataset for our method. Waterbirds uses $E$=200 epochs, RedMeat uses $E$=150. DecoyMNIST uses $E$=19 and does not tune hyperparameters; we use fixed Adam learning rates and set `attention_epoch` by exposure-matching to $\text{WB}_{100}$.

| **Dataset** | `att_ep` | `kl_lambda` | `base_lr` | `classifier_lr` | `lr2_mult` |
|---|---|---|---|---|---|
| $\text{Waterbirds}_{95}$ | 109 | 295.30 | $4.82\times10^{-5}$ | $2.93\times10^{-3}$ | 0.409 |
| $\text{Waterbirds}_{100}$ | 73 | 495.61 | $5.72\times10^{-5}$ | $3.57\times10^{-3}$ | 0.123 |
| DecoyMNIST | 7 | 495.61 | $1.00\times10^{-3}$ | $1.00\times10^{-3}$ | 0.123 |
| RedMeat | 2 | 11.44 | $2.40\times10^{-3}$ | $2.33\times10^{-4}$ | 1.567 |

# C Local sensitivity to R4RR-specific hyperparameters

To assess whether **R4RR** depends on a narrowly tuned operating point, we perform a lightweight local sensitivity study around the validation-selected configuration on $\text{Waterbirds}_{100}$. Starting from the best hyperparameters reported in Table 13, we perturb one R4RR-specific hyperparameter at a time while holding the others fixed. We consider the three method-specific quantities introduced by our two-phase evidence-alignment framework: the alignment start epoch
(`attention_epoch`), the evidence-alignment weight (`kl_lambda`), and the Phase-2 learning-rate multiplier (`lr2_mult`). Each perturbed configuration is evaluated over the same five seeds (0–4) used in the main paper.

Table 14: **Local sensitivity of R4RR-specific hyperparameters on $\text{Waterbirds}_{100}$.** Starting from the validation-selected R4RR configuration, we perturb one hyperparameter at a time while keeping all others fixed. Results are test accuracy (mean±std over 5 seeds). Higher is better.

| **Setting** | $\textbf{Per}_{Group}$ | $\textbf{Worst}_{Group}$ |
|---|---|---|
| `attention_epoch` + 10 | 88.9±0.5 | 83.1±2.6 |
| `attention_epoch` - 10 | 89.0±0.2 | 83.9±2.5 |
| `lr2_mult` × 0.75 | 88.8±0.8 | 83.8±2.0 |
| `lr2_mult` × 1.25 | 89.0±0.3 | 84.9±1.7 |
| `kl_lambda` × 0.75 | 88.4±0.4 | 82.5±1.5 |
| `kl_lambda` × 1.25 | 88.8±0.2 | 85.1±1.6 |
| Best tuned config | 89.1±0.4 | 84.8±1.0 |

Table 14 shows that **R4RR** remains strong under moderate one-at-a-time perturbations of its main method-specific hyperparameters. Across all tested settings, $\text{Per}_{Group}$ varies only within a narrow range (88.4–89.1), and $\text{Worst}_{Group}$ remains similarly stable (82.5–85.1). In particular, modest shifts in the alignment start epoch and the Phase-2 learning-rate multiplier

have only small effects on performance, while changes to `kl_lambda` lead to somewhat larger but still limited variation. Importantly, none of these nearby settings causes a collapse in worst-group accuracy, indicating that the gains of **R4RR** do not rely on an extremely fragile choice of hyperparameters.

Overall, this local sensitivity study supports the view that **R4RR** is reasonably robust around its validation-selected operating point: the tuned setting remains among the strongest, but neighboring configurations continue to deliver high group-balanced performance in the fully biased $\text{Waterbirds}_{100}$ regime. This is encouraging because it suggests that the benefits of evidence alignment arise from the method itself rather than from a razor-thin hyperparameter choice.

# D Teacher ablation on additional datasets

For all the OpenCLIP teacher variants, we use OpenCLIP with LAION-pretrained weights, while keeping the rest of the teacher pipeline unchanged.

Table 15: Teacher ablation on DecoyMNIST and RedMeat dataset: VLM and refinement backbones while keeping the student, loss, and curriculum fixed. Results are accuracy (mean±std). Higher is better.

| **Teacher variant** | **Teacher components** | | **DecoyMNIST** | | **RedMeat** | |
|---|---|---|---|---|---|---|
| | **VLM** | **Refine** | $\textbf{Per}_{Group}$ | $\textbf{Worst}_{Group}$ | $\textbf{Per}_{Group}$ | $\textbf{Worst}_{Group}$ |
| **Baseline (ours)** | CLIP | DINO | 96.6±0.1 | 95.8±1.3 | 73.2±1.0 | 60.9±3.4 |
| OpenCLIP teacher | OpenCLIP | DINO | 96.7±0.4 | 95.6±1.3 | 72.9±1.0 | 60.1±5.1 |
| SigLIP2 teacher | SigLIP2 | DINO | 95.9±0.6 | 94.3±1.4 | 67.9±0.5 | 53.3±7.7 |
| XCiT-DINO refine | CLIP | XCiT-DINO | 96.7±0.5 | 96.0±0.8 | 71.5±0.9 | 54.1±3.3 |

Table 15 extends the teacher-selection study beyond Waterbirds to DecoyMNIST and RedMeat, while keeping the student architecture, loss, and training curriculum fixed. Overall, the results indicate that our method **R4RR** is largely *insensitive* to the specific CLIP-family backbone used for generating teacher maps.

On **DecoyMNIST**, both the baseline CLIP teacher and OpenCLIP teacher achieve essentially identical performance. This suggests that for synthetic tasks where the foreground concept is visually simple and the shortcut is strong but structured, multiple CLIP-family teachers provide sufficiently accurate localization cues for evidence alignment. Swapping the refinement backbone to XCiT-DINO yields similar performance and a small improvement in worst-group accuracy.

On **RedMeat**, the teacher choice has a larger impact. While the baseline CLIP and

OpenCLIP teachers nearly match, SigLIP2 degrades both per-group and worst-group accuracy, and replacing DINO with XCiT-DINO also reduces performance.

The overall trend suggests that stronger teachers yield better worst-group robustness, reinforcing the motivation that **R4RR** benefits from accurate task-relevant localization, but does not require a uniquely tuned teacher to function.

# E Dataset splits

Tables 16, 17, 18 and 19 report the train/validation/test splits used in all experiments. For DecoyMNIST, we create the validation set by taking a stratified 10% slice from the original training split (per digit), and we report the resulting counts below.

Table 16: **DecoyMNIST class-balanced splits (counts per digit).** Validation is a stratified 10% slice from the original training split.

| **Split** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|
| Train | 5331 | 6068 | 5362 | 5518 | 5258 | 4879 | 5326 | 5638 | 5266 | 5354 |
| Validation | 592 | 674 | 596 | 613 | 584 | 542 | 592 | 627 | 585 | 595 |
| Test | 980 | 1135 | 1032 | 1010 | 982 | 892 | 958 | 1028 | 974 | 1009 |

Table 17: **Waterbirds$_{95}$ splits (group counts).** Groups are the four label×background combinations.

| **Split** | **Landbird, Land** | **Landbird, Water** | **Waterbird, Land** | **Waterbird, Water** |
|---|---|---|---|---|
| Train | 3498 | 184 | 56 | 1057 |
| Validation | 467 | 466 | 133 | 133 |
| Test | 2255 | 2255 | 642 | 642 |

Table 18: **Waterbirds$_{100}$ splits (group counts).** Training is fully biased (no counter-biased groups).

| **Split** | **Landbird, Land** | **Landbird, Water** | **Waterbird, Land** | **Waterbird, Water** |
|---|---|---|---|---|
| Train | 3684 | 0 | 0 | 1111 |
| Validation | 465 | 466 | 134 | 134 |
| Test | 2255 | 2255 | 642 | 642 |

Table 19: **RedMeat splits (class counts).** Classes follow [20].

| Split | Baby Back Ribs | Filet Mignon | Pork Chop | Prime Rib | Steak |
|---|---|---|---|---|---|
| Train | 500 | 500 | 500 | 500 | 500 |
| Validation | 250 | 250 | 250 | 250 | 250 |
| Test | 250 | 250 | 250 | 250 | 250 |

Table 20: **Optuna-optimized GALS variant sweep.** Test accuracy (mean±std over 5 runs) for multiple GALS instantiations, each tuned independently with 50 Optuna trials on the validation set. We use the best-performing GALS variant per dataset in the main comparisons.

| | $\textbf{Waterbirds}_{95}$ | | $\textbf{Waterbirds}_{100}$ | | RedMeat | |
|---|---|---|---|---|---|---|
| **GALS variant** | $\textbf{Per}_{Group}$ | $\textbf{Worst}_{Group}$ | $\textbf{Per}_{Group}$ | $\textbf{Worst}_{Group}$ | **Test Acc.** | **Worst Class** |
| GALS RRR (CLIP RN50) | $86.8 \pm 0.4$ | $69.2 \pm 2.2$ | $73.3 \pm 2.0$ | $38.1 \pm 6.3$ | $71.9 \pm 0.3$ | $57.0 \pm 1.5$ |
| GALS RRR (CLIP ViT) | $89.1 \pm 0.4$ | $76.0 \pm 1.9$ | $79.9 \pm 2.1$ | $57.2 \pm 5.5$ | $71.5 \pm 0.5$ | $54.6 \pm 3.8$ |
| GALS ABN (CLIP ViT) | $86.1 \pm 2.2$ | $68.2 \pm 7.5$ | $69.7 \pm 2.7$ | $34.2 \pm 6.4$ | $70.1 \pm 1.4$ | $51.4 \pm 8.4$ |
| GALS Grad-CAM (CLIP ViT) | $84.7 \pm 0.1$ | $68.8 \pm 0.7$ | $71.7 \pm 3.4$ | $39.1 \pm 4.5$ | $70.4 \pm 1.0$ | $51.7 \pm 5.6$ |

# F Additional GALS Optimization: Exhaustive Variant Selection

To ensure our comparisons against GALS [20] are not confounded by suboptimal configuration choices, we additionally performed a focused sweep over several common GALS instantiations and explanation mechanisms. For each dataset, we tuned each GALS variant independently using the same validation-based Optuna protocol as in the main paper (50 trials per configuration), and report mean±std over 5 seeds for the best trial within each variant.

We considered (i) the CLIP backbone used for grounding (RN50 vs. ViT), and (ii) the attention supervision mechanism used within the GALS framework (RRR-style gradient regularization, ABN-style attention branch, and a Grad-CAM based variant). In all main tables, we report the strongest GALS result per dataset from this search, so that our method is compared against the best-performing version of GALS we found rather than a single arbitrary instantiation.

Table 20 shows that, even after independently tuning multiple plausible GALS variants with the same 50-trial Optuna budget, the relative ordering is stable: the best-performing GALS configuration depends on the dataset, but none closes the gap to our method in the hardest spurious-correlation regimes. In particular, CLIP ViT improves over RN50 on Waterbirds (especially $\text{Waterbirds}_{100}$), while RN50 is slightly stronger on RedMeat under

worst-class performance. Based on these results, we always select the best GALS variant per dataset for the main-paper comparisons, ensuring that reported gains are not an artifact of under-tuning or an unfavorable GALS instantiation.

# Detailed Acknowledgements

This thesis followed a nonstandard path relative to the typical Brown CS undergraduate honors process. The research began during the summer at the RIT REU in Trustworthy AI and was later continued and expanded into this Brown honors thesis.

I am grateful to Dr. Dipkamal Bhusal (RIT), who served as my primary advisor on this project, and to Professor Nidhi Rastogi (RIT) for continued advising and key project guidance throughout this work. I also thank Professor Daniel Ritchie (Brown University) for serving as my Brown faculty advisor and for supporting the Brown-side honors process and project development.

R. Yang and N. Rastogi were partially supported by NSF Award #2447631.

# Bibliography


[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer, 2014.

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[4] Alaaeldin El-Nouby, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *arXiv preprint arXiv:2106.09681*, 2021.

[5] Mohamed Elhoseiny, Babak Saleh, and Ahmed Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions. In *Proceedings of the IEEE international conference on computer vision*, pages 2584–2591, 2013.

[6] Gabriel Erion, Joseph D Janizek, Pascal Sturmfels, Scott M Lundberg, and Su-In Lee. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature machine intelligence*, 3(7):620–631, 2021.

[7] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. *Advances in neural information processing systems*, 26, 2013.

[8] Hiroshi Fukui, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. Attention branch network: Learning of attention mechanism for visual explanation. In

*Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10705–10714, 2019.

[9] Yuyang Gao, Tong Steven Sun, Guangji Bai, Siyi Gu, Sungsoo Ray Hong, and Zhao Liang. Res: A robust framework for guiding visual explanation. In *proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 432–442, 2022.

[10] Yuyang Gao, Tong Steven Sun, Liang Zhao, and Sungsoo Ray Hong. Aligning eyes between humans and deep neural network through interactive attention alignment. *Proceedings of the ACM on Human-Computer Interaction*, 6(CSCW2):1–28, 2022.

[11] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[12] Avani Gupta, Saurabh Saini, and PJ Narayanan. Concept distillation: leveraging human-centered explanations for model improvement. *Advances in Neural Information Processing Systems*, 36:63724–63737, 2023.

[13] 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.

[14] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021. If you use this software, please cite it as below.

[15] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020.

[16] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):453–465, 2013.

[17] Yannick Le Cacheux, Hervé Le Borgne, and Michel Crucianu. Using sentences as semantic representations in large scale zero-shot learning. In *European Conference on Computer Vision*, pages 641–645. Springer, 2020.

[18] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

[19] Yuqi Lin, Minghao Chen, Wenxiao Wang, Boxi Wu, Ke Li, Binbin Lin, Haifeng Liu, and Xiaofei He. Clip is also an efficient segmenter: A text-driven approach for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15305–15314, 2023.

[20] Suzanne Petryk, Lisa Dunlap, Keyan Nasseri, Joseph Gonzalez, Trevor Darrell, and Anna Rohrbach. On guiding visual attention with language specification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18092–18102, 2022.

[21] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

[22] Shikai Qiu, Andres Potapczynski, Pavel Izmailov, and Andrew Gordon Wilson. Simple and fast group robustness by automatic feature reweighting. In *International Conference on Machine Learning*, pages 28448–28467. PMLR, 2023.

[23] 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*, pages 8748–8763. PmLR, 2021.

[24] Laura Rieger, Chandan Singh, William Murdoch, and Bin Yu. Interpretations are useful: penalizing explanations to align neural networks with prior knowledge. In *International conference on machine learning*, pages 8116–8126. PMLR, 2020.

[25] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717*, 2017.

[26] 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.

[27] Patrick Schramowski, Wolfgang Stammer, Stefano Teso, Anna Brugger, Franziska Herbert, Xiaoting Shao, Hans-Georg Luigs, Anne-Katrin Mahlein, and Kristian Kersting. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2(8):476–486, 2020.

[28] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[29] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

[30] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, Serge Belongie, et al. The caltech-ucsd birds-200-2011 dataset. Technical report, Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[31] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. *arXiv preprint arXiv:2205.15480*, 2022.

[32] Bingfeng Zhang, Siyue Yu, Yunchao Wei, Yao Zhao, and Jimin Xiao. Frozen clip: A strong backbone for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3796–3806, 2024.

[33] Bingfeng Zhang, Siyue Yu, Jimin Xiao, Yunchao Wei, and Yao Zhao. Frozen clip-dino: A strong backbone for weakly supervised semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[34] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102, 2018.

[35] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[36] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.