Supplementary Material

# Table of Contents


A. Detailed Related Works 13
A.1. Image-Level OOD Detection 13
A.2. Object-Level OOD Detection 14

B. Dataset Information 15
B.1. Object-level OOD Detection Datasets 15
B.2. Image-level OOD Detection Datasets 16

C. Implementation Details 16

D. Supplementary Results 17
D.1. Heatmap Visualization of Our Weights $\mu_{i,(x)}\beta^{\text{img}}_{i,(x)}$ for Background Image 17
D.2. Heatmap Visualization of Our Weights $\mu_{i,x}\beta^{\text{img}}_{i,x}$ for Target Object Image 17
D.3. Heatmap Visualization of Our *Inter*-Text Attention $w^{\text{text}}_{x',x}$ 18
D.4. ID Classification Accuracy 18
D.5. The Impact of the Object Detector 19
D.6. Computational Complexity and Memory Usage Analysis 20
D.7. Usage of Actual Objects in Background for Inference 21
D.8. Ablation Study on the Number of Fine-tuning Samples (shots) 21
D.9. Ablation Study on Different Backbones of CLIP Image Encoder 22
D.10. Discussions of Component Designs 23

# A. Detailed Related Works

## A.1. Image-Level OOD Detection

**Traditional Image-Level OOD Detection.** Early methods in image-level OOD detection primarily relied on backbones based on traditional deep neural networks, such as Convolutional Neural Networks (CNNs) trained on the entire in-distribution (ID) data to identify out-of-distribution (OOD) samples. These methods can be broadly categorized into several paradigms. *Confidence-based methods* utilize the maximum softmax probability [14] or maximum logit scores [15] as indicators of sample normality, with energy-based approaches [27] demonstrating improved robustness against overconfidence. Recent advances include logit normalization [44] and generalized entropy methods [28] that better calibrate model predictions. *Distance-based approaches* measure the distance between test samples and ID class prototypes in feature space, with the Mahalanobis distance method [22] serving as a foundational approach. K-nearest neighbor (KNN) methods [41] have gained prominence for their distribution-free nature, while hyperspherical embeddings [32] and mixture of prototypes [29] provide more flexible density estimation. *Reconstruction-based methods* leverage generative models to detect OOD samples, with recent work exploring masked image modeling [23] and diffusion models [12, 13] for improved reconstruction quality. *Self-supervised and contrastive learning* approaches [40, 42] learn representations that naturally separate ID from OOD data by optimizing for class discrimination, with supervised contrastive learning proving particularly effective when combined with distance-based methods. Additionally, some methods have explored the use of auxiliary synthetic outlier samples [9] and ensemble strategies [52] to enhance OOD detection performance. While these traditional methods have demonstrated good performance, they typically require training on the entire ID dataset and often struggle with hard OOD detection cases, as CNNs are significantly less capable than vision-language models (VLMs).

**Zero-Shot VLM-Based Image-Level OOD Detection.** The advent of VLMs, particularly CLIP [38], has revolutionized OOD detection by enabling zero-shot transfer capabilities. CLIP's joint vision–language embedding space, pre-trained on hundreds of millions of image–text pairs, provides rich semantic representations that can be leveraged for OOD detection without task-specific training. Multiple recent works have explored this paradigm [11]. MCM [31] proposes Maximum Concept Matching, a zero-shot OOD detection method that characterizes OOD uncertainty by measuring the softmax-scaled cosine similarity between visual features and textual concept prototypes of ID classes in CLIP's joint embedding space. GL-MCM [34] extends this by incorporating both global and local features. NegLabel [18] proposes using negative labels to better calibrate OOD scores. Despite their impressive zero-shot capabilities, these methods often underperform compared to methods that can adapt to specific ID distributions through few-shot learning.

**Few-Shot VLM-Based Image-Level OOD Detection.** To bridge the gap between zero-shot generalization and task-specific performance, recent works have focused on adapting VLMs using only a few labeled ID samples. This few-shot paradigm is particularly relevant for practical applications of OOD detection, where a limited number of labeled ID data is available. CoOp [59] pioneered this direction by introducing learnable prompt tokens that are optimized using few-shot ID samples, allowing the text encoder to better align with the specific ID classes. LoCoOp [33] extends CoOp by introducing local regularization for OOD detection: it treats ID-irrelevant local CLIP features, such as nuisance background regions, as OOD-like features during training and pushes them away from ID class text embeddings, thereby improving ID and OOD separation. PLOT [2] learns multiple comprehensive prompts to describe diverse characteristics of categories and applies optimal transport to align local visual features with multiple textual prompts via the Sinkhorn algorithm, achieving fine-grained cross-modal matching. More recent advances have incorporated additional mechanisms for improved OOD detection performance. GalLoP [21] learns multiple diverse prompts leveraging both global and local visual features, using a sparsity strategy for local feature selection and a prompt dropout technique to encourage diversity among prompts. DPM-T [56] proposes Dual-Pattern Matching that leverages both textual and visual ID patterns for OOD detection, storing ID class-wise text features as the textual pattern and aggregated ID visual information as the visual pattern, and further extends this with lightweight prompt learning adaptation. SCT [54] proposes Self-Calibrated Tuning, which introduces modulating factors into the learning objective to dynamically adjust the importance of OOD regularization based on prediction uncertainty during prompt tuning, mitigating the negative effect of spurious OOD features extracted from ID data. LSN [36] and NegPrompt [24] both explicitly explore the trainable negative and positive text prompts, which helps to further boost the OOD detection performance. ID-like [1] introduces a novel perspective by training the model to recognize OOD samples that appear similar to ID classes, thereby explicitly modeling the decision boundary.

While these few-shot methods have achieved significant improvements over zero-shot approaches, they still only focus on image-level features and do not explicitly address challenges that arise when directly adapting these techniques to object-level OOD detection, where multiple objects may coexist in a single image with varying levels of context information.

## A.2. Object-Level OOD Detection

Object-level OOD detection extends the formulation of image-level OOD detection to the more challenging task of identifying unknown objects within complex scenes where multiple objects coexist. Unlike image-level detection, which provides a single decision for the entire image, object-level OOD detection must simultaneously localize objects and determine whether each detected object belongs to ID or OOD. This task is particularly crucial for safety-critical applications such as autonomous driving and robotics, where encountering unknown objects is inevitable in open-world environments.

**Traditional Object-Level OOD Detection.** Early methods for object-level OOD detection are primarily built upon backbone models based on CNNs, adapting techniques from image-level OOD detection while addressing unique challenges posed by the object detection framework. These approaches can be broadly categorized based on their training strategies and uncertainty modeling techniques.

One prominent direction focuses on synthesizing virtual outliers to provide supervision signals for unknown objects during training. VOS [8] pioneered this approach by adaptively sampling virtual outliers from low-likelihood regions of the class-conditional distribution in feature space, introducing an unknown-aware training objective that contrastively shapes the uncertainty space between ID and synthesized OOD data. Building upon distance-based methods, SIREN [7] proposes a framework that shapes representations into von Mises-Fisher (vMF) distributions on the unit hypersphere, enabling both parametric and non-parametric OOD scoring functions, which demonstrates strong performance across multiple benchmarks.

Another line of work explores leveraging additional model components or features to improve OOD detection performance. SAFE [45] introduces a post-hoc approach that identifies sensitivity-aware features from residual convolutional layers with batch normalization, training a multilayer perceptron on the surrogate task of distinguishing adversarially perturbed from clean ID examples without requiring actual OOD training data or retraining of the base detector. SR-VAE [46] employs a Structure-Enhanced Recurrent Variational AutoEncoder with dedicated recurrent VAE branches, utilizing Laplacian of Gaussian operators to enhance structural information for improved object localization in object-level OOD detection.

More recent approaches focus on addressing the fundamental challenge of lacking OOD objects' information through innovative information modeling strategies. TIB [47] proposes a Two-Stream Information Bottleneck consisting of a standard information bottleneck for disentangling instance representations and a dedicated Reverse Information Bottleneck (RIB) to synthesize simulative OOD features, mitigating the impact of absent OOD samples. The PCA-driven method [50] utilizes Principal Component Analysis (PCA) to extract dynamic prototypes from residual principal components, employing contrastive learning to enlarge the semantic gap between features of OOD and ID samples. DFDD [49] explores deep feature deblurring via diffusion for robust object-level OOD detection. Most recently, WFS [48] advances the field by proposing a World Feature Simulation, with components for multi-level perception, memory, and imagination or unknown-feature generation.

Beyond the above strategies, prototype-based and context-leveraging methods have emerged as complementary directions for object-level OOD detection. Proto-OOD [3] introduces a prototype-based detection framework that leverages contrastive loss to enhance the representativeness of per-class prototypes and employs a similarity module to evaluate how closely each detected object's features align with these prototypes. A negative embedding generator is used during training to synthesize pseudo-OOD features for the similarity module, enabling effective ID and OOD discrimination without requiring real OOD data. Additionally, Situation Monitor [37] proposes a diversity-driven zero-shot OOD detection approach specifically targeting transformer-based object detectors. By integrating a diversity loss into a budding ensemble architecture, the method captures prediction disagreement between ensemble branches as an uncertainty signal, thereby identifying far-OOD samples while minimizing false positives on near-OOD instances. This ensemble-based uncertainty estimation provides a complementary perspective to single-model OOD scoring.

Despite their effectiveness, these traditional methods require full training on the entire ID dataset, often involving complex modifications to the detector architecture or training procedures. Moreover, they are fundamentally limited by the closed-set assumption and the restricted capability of CNNs compared with VLMs.

**Generative-Consistency Object-Level OOD Detection.** A recent and promising direction leverages pre-trained generative models to detect OOD objects by measuring reconstruction or synthesis consistency rather than relying solely on discriminative representations. RONIN [35] proposes a zero-shot framework that exploits contextual scene information through class-conditioned inpainting using an off-the-shelf text-to-image diffusion model (e.g., Stable Diffusion). Given a detected object's bounding box and the predicted ID class label, the object is removed and regenerated via class-conditioned inpainting. For ID objects, the inpainted result closely resembles the original, whereas for OOD objects, the mismatch is significant. This reconstruction error thus serves as a discriminative OOD indicator. Notably, RONIN requires no additional training on ID data, operating entirely in a zero-shot manner by leveraging the generative prior's understanding of object-context relationships.

In a complementary direction, SyncOOD [26] addresses

the chronic shortage of OOD training data by proposing an automated data curation pipeline that uses multiple foundation models in concert. Specifically, it employs a large language model to discover semantically novel object concepts, then uses Stable Diffusion to inpaint these novel objects into existing ID scenes, and finally applies the Segment Anything Model (SAM) [19] for annotation and visual filtering. The resulting synthetic OOD data is used to train a lightweight, plug-and-play binary OOD classifier on top of a frozen object detector, enabling effective ID/OOD decision boundary optimization with minimal synthetic data.

**VLM-Based Object-Level OOD Detection.** The emergence of VLMs has opened new possibilities for object-level OOD detection. RUNA [55] represents the *first* and *only* work to adapt CLIP, one of the most popular VLMs, for object-level OOD detection through few-shot learning. The method introduces a regional uncertainty alignment framework that fine-tunes a regional image encoder with only 10-shot ID samples, leveraging the rich semantic knowledge from CLIP's vision–language pre-training. By adapting CLIP's multi-modal representations to the object-level detection task, RUNA achieves substantial improvements over traditional methods while requiring significantly less training data.

Despite these advancements, RUNA still exhibits multiple fundamental drawbacks. The first one is due to the exclusive reliance on the `[CLS]` embedding of the CLIP image encoder. To understand it more concretely, consider RUNA's regional encoder design. For each detected object, RUNA crops the bounding box region and processes it through CLIP's image encoder, extracting only the `[CLS]` embedding as the object representation [55]. While `[CLS]` embedding provides a summary of the object, it only aggregates information across all locations, thereby discarding the fine-grained visual details that may be encoded in patch output embeddings. This aggregation becomes particularly problematic for challenging cases: small objects where limited pixels provide insufficient information after cropping and resizing; heavily occluded objects where only partial visual evidence remains visible; and visually similar object pairs (e.g., bus vs. truck) where subtle structural differences, precisely the information lost in `[CLS]` embedding, determine the correct OOD decision for objects. The Vision Transformer (ViT) [6] in CLIP image encoder explicitly preserves these fine-grained visual details in its patch output embeddings, yet RUNA's exclusive reliance on the `[CLS]` embedding leaves these rich representations untouched, limiting its ability to handle hard OOD detection cases.

The second limitation manifests in RUNA's strategy for background processing. To incorporate contextual information, RUNA applies a Gaussian blur to the entire background in a unified way, then extracts the `[CLS]` embedding based on this blurred image as the background representation [55]. However, this approach conflates a challenge: identifying *which* background regions contain relevant spurious cues to the object. By extracting only the `[CLS]` embedding, RUNA yet again takes a simple summary of the entire background with a Gaussian blur. This approach fails when semantically relevant context appears only in certain regions of the background: a forest background that distinguishes wolves from huskies, or urban infrastructure that separates buses from trucks.

## B. Dataset Information

In this section, we provide comprehensive information on all datasets used in our experiments for both object- and image-level OOD detection.

### B.1. Object-level OOD Detection Datasets

For object-level OOD detection, we follow the popular experimental setup in RUNA [55]. We provide details of each dataset used in this task in the subsequent paragraphs.

**PASCAL-VOC.** The PASCAL Visual Object Classes (VOC) [10] dataset is a widely-used benchmark for object detection tasks. It contains 20 object categories, including common objects such as person, car, bicycle, and various animals. The combined dataset contains approximately 16,551 training images and 4,952 validation images. Each image is annotated with bounding boxes for multiple objects, making it suitable for training and evaluating object-level OOD detection methods.

**BDD-100k.** The Berkeley Deep Drive dataset (BDD-100k) [53] is a large-scale, diverse driving dataset designed for heterogeneous multitask learning. It contains 100,000 video sequences with diverse driving scenarios captured under various weather conditions, times of day, and locations across the United States. For object detection, BDD-100k includes 10 object categories commonly encountered in autonomous driving scenarios: car, bus, person, bike, truck, motor, train, rider, traffic light, and traffic sign. The dataset provides 69,853 images for training and 10,000 images for validation. The diversity of BDD-100k, with varying lighting conditions, occlusions, and object scales, makes it ideal for object-level OOD detection.

**MS-COCO.** The Microsoft Common Objects in Context (MS-COCO) [25] dataset is a large-scale object detection, segmentation, and captioning dataset containing 80 object categories. The dataset comprises 118,287 training images, 5,000 validation images, and over 40,000 test images. We adopt the dataset setup from RUNA [55] and VOS [8], which carefully select object categories from MS-COCO that do not overlap with the corresponding ID dataset (PASCAL-VOC or BDD-100k) to serve as OOD samples. Specifically, when PASCAL-VOC is used as the ID dataset, we exclude the 20 overlapping categories from MS-COCO and use the remaining 60 categories as OOD. When BDD-100k is the

ID dataset, we exclude the 10 overlapping categories and use the remaining 70 categories as OOD. This ensures a clear separation between ID and OOD distributions during evaluation.

**OpenImages.** OpenImages [20] is a large-scale dataset containing approximately 9 million images annotated with image-level labels, object bounding boxes, visual relationships, and localized narratives. The object detection subset contains 600 object categories with 1.9 million images for training and 125,000 images for validation. Similar to MS-COCO, we adopt the dataset setup from RUNA [55] and VOS [8], and select categories from OpenImages that do not overlap with the ID categories. When PASCAL-VOC is used as ID, we exclude the 20 overlapping categories from OpenImages' 600 categories. When BDD-100k is used as ID, we exclude the 10 overlapping categories and use the remaining categories as OOD. The diversity and large number of categories in OpenImages provide a comprehensive evaluation of the model's ability to detect various OOD objects.

### B.2. Image-level OOD Detection Datasets

For image-level OOD detection, we follow the popular experimental setup as in ID-like [1], using ImageNet-1k as the ID dataset and evaluating on multiple OOD datasets.

**ImageNet-1k.** ImageNet [5] is one of the most widely used large-scale image classification datasets, containing 1,000 object categories with approximately 1.28 million training images and 50,000 validation images. The dataset covers a diverse range of object categories, including animals, vehicles, household objects, and natural scenes. The large number of categories and diverse visual content in ImageNet-1k make it an ideal benchmark for evaluating OOD detection methods in realistic scenarios.

Following the protocol established in [1, 16, 31], we use four OOD datasets where the categories do not overlap with ImageNet-1k classes. Below, we provide detailed descriptions of each OOD dataset.

**iNaturalist.** The iNaturalist dataset [43] is a large-scale species classification dataset containing images of various plants and animals collected from the natural world. It includes 13 super-categories (such as Plantae, Insecta, Aves, Mammalia, etc.) with 5,089 fine-grained sub-categories. For OOD evaluation, following [1, 16], we use a carefully curated subset containing 110 species categories that do not overlap with ImageNet-1k. These categories primarily consist of plant species and less common animal species that are not present in ImageNet-1k, providing a semantically related but distributionally different test set for evaluating OOD detection performance. The total number of images in this OOD set is approximately 10,000.

**PLACES.** Places365 (PLACES) [57] is a large-scale scene recognition dataset containing 1.8 million training images from 365 scene categories. The categories cover a wide range of indoor, urban, and natural environments, including offices, restaurants, streets, forests, and beaches. Unlike ImageNet-1k, which focuses primarily on objects, PLACES emphasizes scene-level understanding, making it a challenging OOD dataset. Following [1, 16], we use a subset of 50 scene categories that do not overlap with ImageNet-1k's object categories. This subset contains approximately 10,000 images and represents a significant semantic shift from object-centric to scene-centric images, testing the model's ability to detect OOD samples with different visual characteristics.

**TEXTURE.** The Describable Textures Dataset (TEXTURE) [4] contains images of textures and patterns in the wild. It consists of 5,640 images organized into 47 texture categories such as banded, bubbly, chequered, flecked, marbled, and zigzagged. Unlike object-centric datasets, TEXTURE focuses on local texture patterns rather than global object semantics. Since none of the texture categories overlap with ImageNet-1k's object categories, we use the entire dataset as the OOD set for evaluation. The texture-based nature of this dataset provides a unique challenge for OOD detection, as the model must distinguish between object-centric ID images and texture-centric OOD images.

**SUN.** The Scene UNderstanding (SUN) dataset [51] is a comprehensive scene recognition dataset containing 899 scene categories with over 130,000 images. The categories encompass a wide variety of indoor, urban, and natural environments, with some categories including human activities. Following [1, 16], we use a subset of 50 categories that do not overlap with ImageNet-1k's object categories. This subset contains approximately 10,000 images. Similar to PLACES, SUN represents a scene-centric distribution that differs from ImageNet-1k's object-centric nature, but with a more diverse range of scene types, including scenes with and without human presence, which provides an additional challenge for OOD detection methods.

Table 5 summarizes the key statistics of all datasets used in our experiments of object- and image-level OOD detection tasks, including the number of categories and images available for both training and testing or validation. Please note that for VLM-based methods, the total number of training samples is *not* the real number of samples used in the few-shot fine-tuning stage, but the size of the pool where the few-shot samples are randomly drawn from. This provides a comprehensive overview of the scale and diversity of our experimental evaluation.

## C. Implementation Details

In this section, we provide detailed information on our implementations, encompassing both hardware and software components. We implement our framework using PyTorch 2.8.0 in Python 3.11. By default, all experiments are executed on a four-GPU NVIDIA RTX 5090 workstation. To assess the efficiency and scalability of our method in a more

Table 5. Summary statistics of datasets used in our experiments. Note that for object-level OOD detection, the number of OOD categories and samples varies depending on whether PASCAL-VOC or BDD-100k is used as the ID dataset. "X/Y" numbers for MS-COCO and OpenImages datasets denote the number of categories/samples used when PASCAL-VOC and BDD-100k are ID datasets, respectively. Also, the number of training images for either object-level or image-level OOD detection is *not* the real number of few-shot samples used in the few-shot fine-tuning stage for VLM-based methods, but the total size of the pool from which the few-shot fine-tuning samples are randomly drawn.

| Dataset | Role | Categories | Train | Test/Val |
|---|---|---|---|---|
| *Object-level OOD Detection* | | | | |
| PASCAL-VOC | ID | 20 | 16,551 | 4,952 (Val) |
| BDD-100k | ID | 10 | 69,853 | 10,000 (Val) |
| MS-COCO | OOD | 60/70 | - | 930/1,761 (Test) |
| OpenImages | OOD | 580/590 | - | 1,880/1,761 (Test) |
| *Image-level OOD Detection* | | | | |
| ImageNet-1k | ID | 1,000 | 1,281,167 | 50,000 (Val) |
| iNaturalist | OOD | 110 | - | 10,000 (Test) |
| PLACES | OOD | 50 | - | 10,000 (Test) |
| TEXTURE | OOD | 47 | - | 5,640 (Test) |
| SUN | OOD | 50 | - | 10,000 (Test) |

accessible hardware environment, we additionally evaluate it using a GPU workstation with a single NVIDIA RTX 3090 GPU. For the few-shot learning setup in object-level OOD detection, we adopt the settings in RUNA [55] and use 10-shot for a fair comparison. For the few-shot learning setup in image-level OOD detection, we adopt the settings in ID-like [1] and use 16-shot for a fair comparison. We also use OWLv2 [30] as the object detector for all methods (including competing methods and our proposed method). To optimize the CNNs and MLPs in the combiner module of our framework, we use a batch size of 256 with 100 epochs based on the AdamW optimizer at a learning rate of $5 \times 10^{-6}$ for few-shot fine-tuning on object-level OOD detection task, and a batch size of 128 with 100 epochs based on the AdamW optimizer at a learning rate of $5 \times 10^{-6}$ for few-shot fine-tuning on image-level OOD detection task.

# D. Supplementary Results

## D.1. Heatmap Visualization of Our Weights $\mu_{i,(x)}\beta^{\text{img}}_{i,(x)}$ for Background Image

As illustrated in Section 3.3, we introduce two sets of weights: *inter*-image attention $\{\beta^{\text{img}}_{i,(x)}\}^{N}_{i=0}$ and text–vision alignment $\{\mu_{i,(x)}\}^{N}_{i=0}$, which capture two complementary spurious cues in the background image: (i) the *inter*-visual relevance between representation of target object image $I_x$ and the visual features of background image $I_{(x)}$, and (ii) the alignment between the visual appearance and textual descriptions of background. These background features contain potential spurious cues related to the target object $I_x$, which provide informative context for both object-level and image-level OOD detection.

In this section, we present additional qualitative analysis on our proposed weights $\mu_{i,(x)}\beta^{\text{img}}_{i,(x)}$ for the background in our method via heatmap visualizations based on object-level OOD detection tasks, as shown in Figure 3. Specifically, we provide heatmap visualization for each individual set of weights: *inter*-image attention $\beta^{\text{img}}_{i,(x)}$ and text–vision alignment $\mu_{i,(x)}$, as well as the combined weights $\mu_{i,(x)}\beta^{\text{img}}_{i,(x)}$ in background image. It could be observed that weights $\beta^{\text{img}}_{i,(x)}$ and $\mu_{i,(x)}$ are effectively assigned to higher values for different patches of the background image that contains potential spurious cues, which provide complementary contextual information that contributes to capture more potential spurious cues related to target object when used together (e.g., container and other fruits/vegetables in background for potato as the target object). Also, it shows that both sets of weights $\beta^{\text{img}}_{i,(x)}$ and $\mu_{i,(x)}$ would be low for the masked out region, which is the original location of the target object. Therefore, the patch embeddings of the background image combined by our weight $\mu_{i,(x)}\beta^{\text{img}}_{i,(x)}$ supply useful spurious cues from the background for the target object and thereby contribute to a better OOD detection performance.

## D.2. Heatmap Visualization of Our Weights $\mu_{i,x}\beta^{\text{img}}_{i,x}$ for Target Object Image

As illustrated in Section 3.2, we introduce two sets of weights: *intra*-image attention $\{\beta^{\text{img}}_{i,x}\}^{N}_{i=0}$ and text–vision alignment $\{\mu_{i,x}\}^{N}_{i=0}$, which capture two complementary fine-grained visual cues in the target object image: (i) the *intra*-visual relevance between the holistic representation and local spatial details of $I_x$, and (ii) the alignment between the visual appearance and textual description of the target object $x$. These target object features contain fine-grained representations of target object $I_x$, which is essential for both object-level and image-level OOD detection.

In this section, we provide additional qualitative analysis on our proposed weight $\mu_{i,x}\beta^{\text{img}}_{i,x}$ for the target object image in our method via heatmap visualizations based on object-level OOD detection tasks, as shown in Figure 4. Specifically, we provide heatmap visualization for each individual set of weights: *intra*-image attention $\beta^{\text{img}}_{i,x}$ and text–vision alignment $\mu_{i,x}$, as well as the combined weights $\mu_{i,x}\beta^{\text{img}}_{i,x}$ in target object image. It could be observed that both $\beta^{\text{img}}_{i,x}$ and $\mu_{i,x}$ receive higher values for different patches of the target object image that contains fine-grained visual details, which provide complementary contextual information that contributes to capture more fine-grained visual semantics of target object when used together (e.g., neck (with dominant skin patterns) and antlers (with unique shape and structure) of giraffe). Therefore, the patch embeddings of the target

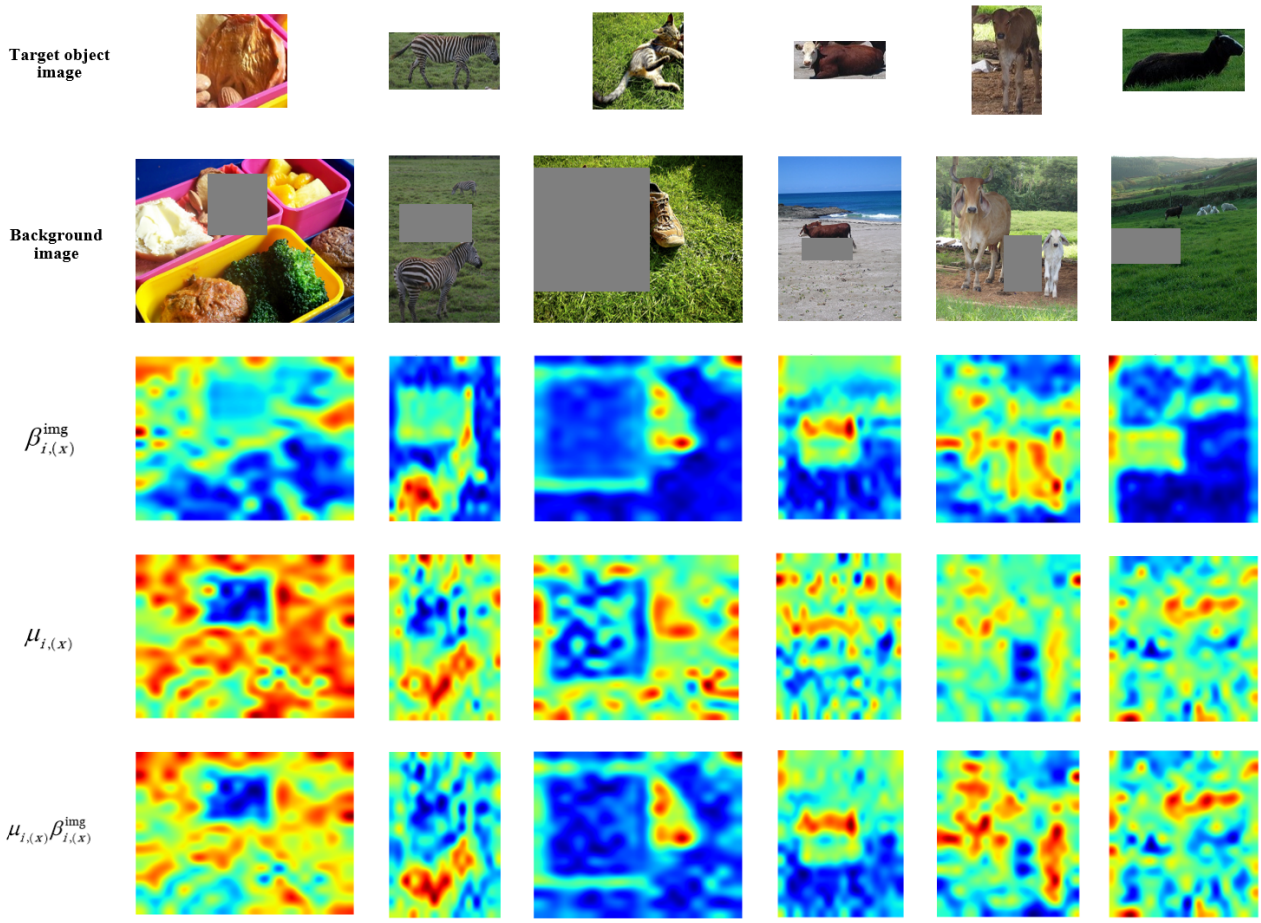


Figure 3. Heatmap visualizations of our weights $\mu_{i,(x)}\beta^{\text{img}}_{i,(x)}$ on background image. In each column, we display the target object image, the corresponding background image, heatmap visualization of our proposed *inter*-image attention weights $\beta^{\text{img}}_{i,(x)}$, heatmap visualization of our proposed text–vision alignment weights $\mu_{i,(x)}$, and heatmap visualization of our proposed combined weights $\mu_{i,(x)}\beta^{\text{img}}_{i,(x)}$ for each patch embedding of the background image, from top to bottom, respectively.

object image combined by our weights $\mu_{i,x}\beta^{\text{img}}_{i,x}$ supply fine-grained visual semantics for the target object and contribute to a better OOD detection performance.

### D.3. Heatmap Visualization of Our *Inter*-Text Attention $w^{\text{text}}_{x',x}$

In this section, we provide a further analysis of our *inter*-text attention $w^{\text{text}}_{x',x}$ based on object-level OOD detection tasks, shown in Figure 5. It is observed that our *inter*-text attention $w^{\text{text}}_{x',x}$ would naturally assign higher weights to objects in the background image that are semantically closer to the target object (e.g., Person & Cow, Chair & Sofa, TV & Person, etc.), which is especially helpful to combine text embeddings of all objects in background into a single text embedding that captures the holistic textual semantics of background explaining multiple objects based on their relevance to the target objects in textual semantics.

### D.4. ID Classification Accuracy

We also evaluate the classification accuracy on ID data when using our proposed method for the image-level OOD detection task, as shown in Table 6. This provides crucial insights into the balance between ID classification and OOD detection performance. It shows that our method is able to beat GalLoP, the *former* best method in ID classification accuracy, as we specifically introduce weights to optimally combine the `[CLS]` and patch output embeddings of target object image and background image, which help to capture more fine-grained visual details in the target object image and relevant spurious cues in the background image, respectively. Those combined representations provide more contextual information, which is beneficial to the classification performance on the ID dataset.

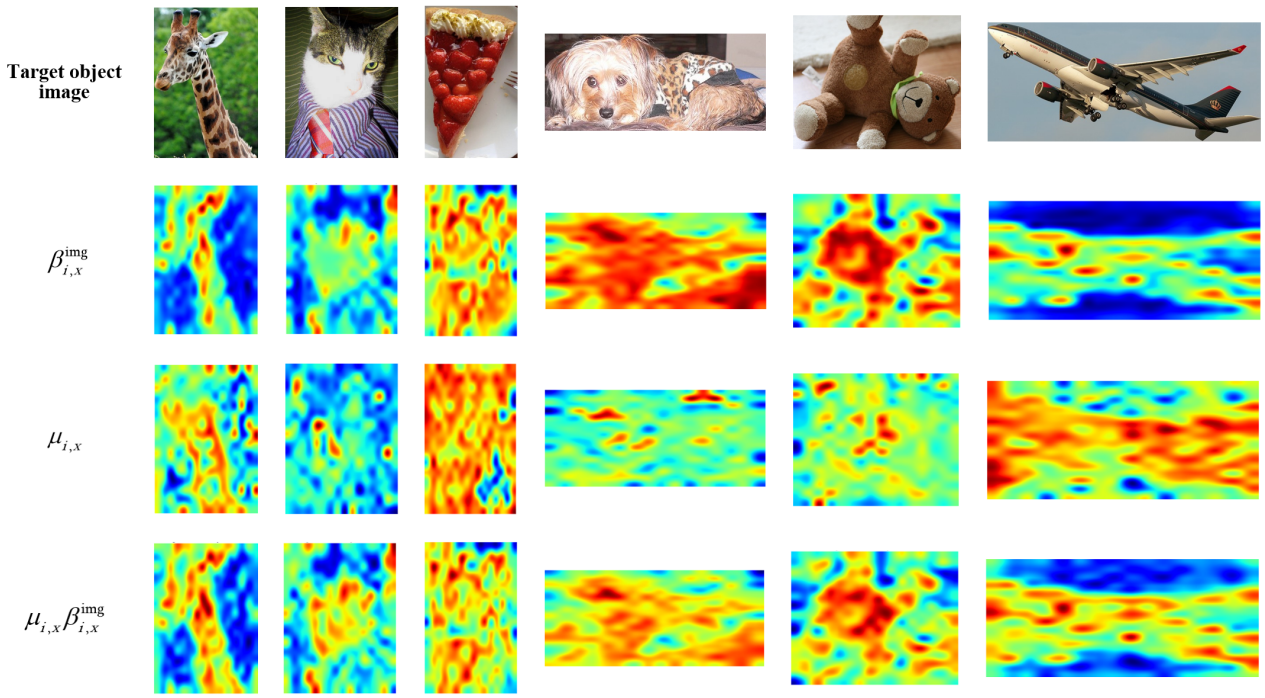


Figure 4. Heatmap visualizations of our weights $\mu_{i,x}\beta_{i,x}^{\text{img}}$ on target object image. In each column, we display the target object image, heatmap visualization of our proposed *intra*-image attention weights $\beta_{i,x}^{\text{img}}$, heatmap visualization of our proposed text–vision alignment weights $\mu_{i,x}$, and the heatmap visualization of our proposed combined weights $\mu_{i,x}\beta_{i,x}^{\text{img}}$ for each patch embedding of the target object image, from top to bottom, respectively.

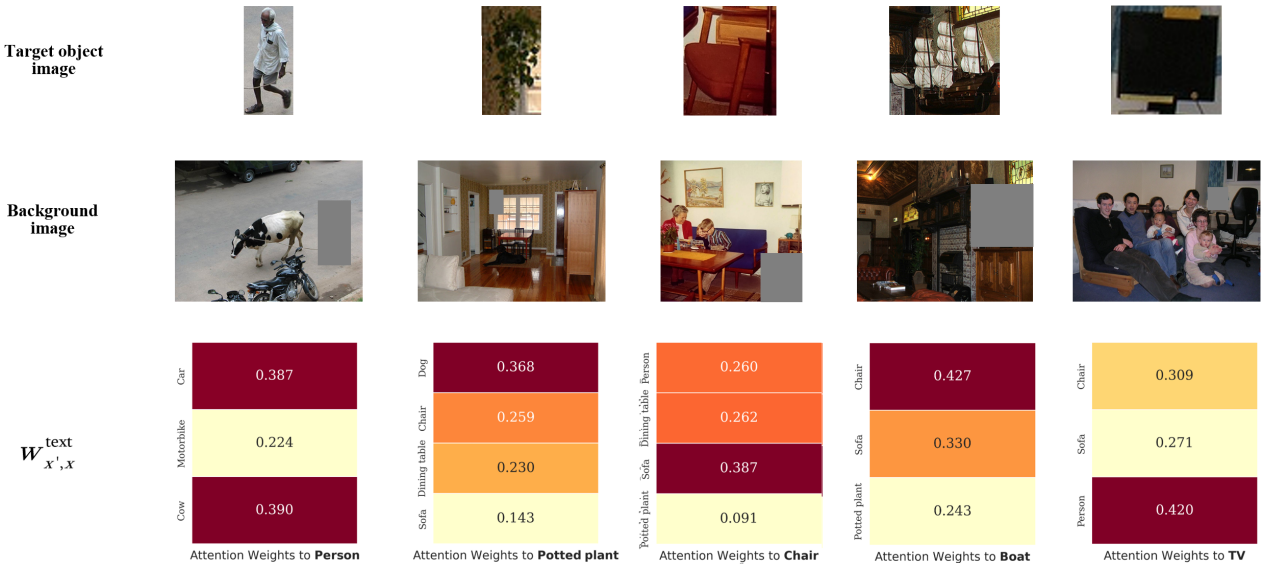


Figure 5. Heatmap visualization of our *inter*-text attention $w_{x',x}^{\text{text}}$. In each column, we show the target object image, the corresponding background image, and a heatmap visualization of our *inter*-text attention $w_{x',x}^{\text{text}}$ for the objects in the background image, from top to bottom, respectively.

### D.5. The Impact of the Object Detector

In the experiments, we use OWLv2 [30] as the object detector for all methods, including both object-level competing

Table 6. Top-1 classification accuracy (↑) on ImageNet-1k (ID dataset). Results with † are taken from [54]. **Bold** and underlined numbers represent best and second-best results, respectively.

| Method | Top-1 Accuracy (↑) on ID dataset |
|---|---|
| CoOp† [59] | 71.93 |
| PLOT [2] | 72.60 |
| LoCoOp† [33] | 71.43 |
| GalLoP [21] | <u>75.10</u> |
| DPM-T [56] | 72.10 |
| SCT† [54] | 71.77 |
| LSN† [36] | 71.93 |
| ID-like† [1] | 71.04 |
| NegPrompt [24] | 72.10 |
| **Ours** | **75.65**±0.19 |

methods and our proposed method, to ensure a fair comparison of performance. In this subsection, we present additional results on the impact of the object detector on image-level OOD detection methods and on the robustness of our proposed method to different types of object detector errors.

**Impact of the object detector on image-level OOD detection methods.** We present the performance comparison of SOTA image-level OOD detection methods with (W/) and without (W/o) the object detector together with our proposed method in Table 7. It could be observed that the object detector only slightly improves the performance of the SOTA image-level baselines, whereas our method still consistently achieves the best performance. This is because our proposed method explicitly models complementary cues from the object and background views by cross-context reasoning.

Table 7. Average AUROC (↑) scores for SOTA Image-level OOD detection methods with (W/) and without (W/o) object detector based on four test cases of image-level OOD detection tasks with ImageNet-1k as ID dataset. **Bold** numbers represent the best results.

| Config. | DPM-T [56] | NegPrompt [24] | SCT [54] | Ours |
|---|---|---|---|---|
| W/o detector | 95.72±0.20 | 94.81±0.27 | 93.12±0.31 | N/A |
| W/ detector | 95.86±0.25 | 94.98±0.27 | 93.25±0.29 | **96.83**±0.15 |

**Robustness of our proposed method to different types of object detector errors.** To assess the robustness of our proposed method to errors from the object detector results, we consider multiple types of errors, including missed objects by randomly dropping detected object boxes, false positives by adding random boxes to the detection results, and localization noise by perturbing detected object box coordinates. Table 8 shows that our proposed method remains robust to different types of object detector errors: our proposed method, with certain object detector errors (10% or 20%), still outperforms SOTA object-level OOD detection methods with perfect object detector output (0%).

Table 8. Average AUROC (↑) scores for our proposed method with different types of object detector errors based on four test cases (BDD-100k→OpenImages, BDD-100k→MS-COCO, PASCAL-VOC→OpenImages, and PASCAL-VOC→MS-COCO) of object-level OOD detection tasks. Percentages indicate the error rate applied in each row.

| Detector error type | Ours (10%) | Ours (20%) | RUNA [55] (0%) | WFS [48] (0%) |
|---|---|---|---|---|
| Missed objects rate | 95.84±0.11 | 95.39±0.18 | 94.55±0.31 | 91.94±0.35 |
| False positive rate | 95.93±0.15 | 95.52±0.24 | 94.55±0.31 | 91.94±0.35 |
| Localization noise rate | 96.05±0.17 | 95.77±0.25 | 94.55±0.31 | 91.94±0.35 |

## D.6. Computational Complexity and Memory Usage Analysis

In this subsection, we present the computational complexity and memory usage analysis of our proposed method in the inference phase, including the scalability with respect to the number of detected objects and inference efficiency comparison with other competing approaches.

**Scalability with respect to the number of detected objects.** Table 9 shows that our proposed method scales efficiently in both computational complexity and memory usage. For memory, our method requires only ∼6 GB for five-object scenes, which is well within typical GPU capacity. Regarding computational complexity, our method requires only ∼111 GFLOPs for five-object scenes. For reference, this translates to ∼24 FPS on a popular RTX 3090 GPU, which is suitable for practical deployment. Furthermore, the inference efficiency of our method can be further improved by leveraging object batching and parallelization.

Table 9. Scalability analysis of our proposed method (with ViT-B/16 for CLIP image encoder) based on computational complexity (GFLOPs) and memory usage (GB) with respect to the number of detected objects. All numbers are measured based on experiments with the batch size set to 1.

| Metric | 1 obj. | 2 obj. | 3 obj. | 4 obj. | 5 obj. |
|---|---|---|---|---|---|
| Computational Complexity (GFLOPs) | 38.2 | 56.4 | 74.6 | 92.8 | 111.0 |
| Memory Usage (GB) | 3.0 | 3.8 | 4.5 | 5.3 | 6.0 |

**Inference efficiency comparison with other competing approaches.** We compare the inference complexity of our method against representative competing approaches for both object- and image-level OOD detection, as summarized in Table 10. For object-level OOD detection, we compare against the VLM-based RUNA [55] and the traditional CNN-based WFS [48]. For image-level OOD detection, we compare against DPM-T [56] and NegPrompt [24]. All VLM-based methods use CLIP (ViT-B/16) as the backbone, and the cost of the OWLv2 object detector is excluded, as it is shared across all VLM-based methods. For traditional methods (WFS), the detector backbone cost is included, as OOD detection is inherently integrated into the detection

pipeline. Object-level results are reported with 3 detected objects per image, which is representative of typical multi-object scenes.

Among object-level methods, WFS incurs the highest computational cost (∼118.5 GFLOPs) due to its heavy Faster R-CNN backbone with ResNet-101 and FPN, which must process the full-resolution image. RUNA is more efficient than WFS, benefiting from the compact CLIP encoder, but its per-object cost (∼17.7 GFLOPs per additional object) is only marginally lower than ours (∼18.2 GFLOPs per additional object), since both methods require a full Image Encoder forward pass for each target object. The additional overhead of our method relative to RUNA stems from the CNN combiners, *intra*-image attention processing, and *inter*-text attention mechanisms, which collectively add only ∼3.6 GFLOPs (∼5.1%) for 3 objects while enabling the substantial AUROC improvements reported in Table 1. Please note that our framework's two identical encoder pairs share all pretrained weights, so no additional model parameters are loaded compared to a single CLIP encoder pair. The marginal memory overhead arises only from intermediate activations (patch embeddings, attention maps, and combiner features) during the forward pass.

For image-level OOD detection, RUNA exhibits an inference efficiency pattern similar to that of object-level OOD detection tasks, which is only slightly lower than ours. However, its image-level OOD detection performance is significantly worse than that of our proposed method (Table 2). As for specialized methods such as NegPrompt and DPM-T, they require only a single CLIP image encoder forward pass and thus have lower computational cost. However, these methods are inherently limited to image-level detection and cannot operate on multi-object scenes. Our method requires approximately 1.8-2.1× the GFLOPs of image-level baselines, because it processes both the target object and its background through separate encoder branches. This overhead is the direct cost of enabling unified object- and image-level OOD detection within a single framework, without prior knowledge of the task type at inference time. Importantly, even at 38.2 GFLOPs for a single-object image, our method translates to ∼62 FPS on a popular RTX 3090 GPU, which remains practical for real-time deployment.

## D.7. Usage of Actual Objects in Background for Inference

In our proposed method, we introduce an efficient strategy that constructs a *single* background text embedding conditioned on each target object in the input image, thereby significantly reducing computational complexity by avoiding the computation of all possible combinations of background objects across the entire ID set (for full details, please see Section 3.5). In this subsection, we present extended results on performance and inference complexity comparisons between our efficient strategy and the exhaustive-search approach (i.e., oracle, the upper bound) in Table 11. It shows that our proposed method achieves nearly identical performance to the oracle upper bound while incurring an over 100× speedup in inference, making it practical for real-world deployment.

Table 10. Inference efficiency comparison of our method with competing approaches for object- and image-level OOD detection tasks. Object-level results are reported with 3 detected objects per image. All VLM-based methods use CLIP (ViT-B/16). Computational complexity is measured in GFLOPs and memory usage in GB (batch size = 1). For VLM-based methods, OWLv2 detection cost is excluded as it is a shared preprocessing step. For WFS, the Faster R-CNN detector cost is included as the OOD module is integrated.

| **Method** | **Type** | Object-level (3 obj.) | | Image-level (1 obj.) | |
|---|---|---|---|---|---|
| | | GFLOPs | Mem. (GB) | GFLOPs | Mem. (GB) |
| *Object-level OOD Detection Methods* | | | | | |
| WFS [48] | Trad. (full train) | 118.5 | 5.6 | N/A | N/A |
| RUNA [55] | VLM (10-shot) | 71.0 | 4.1 | 35.6 | 2.9 |
| *Image-level OOD Detection Methods* | | | | | |
| NegPrompt [24] | VLM (16-shot) | N/A | N/A | 18.4 | 1.8 |
| DPM-T [56] | VLM (16-shot) | N/A | N/A | 21.5 | 2.2 |
| *Unified Object- and Image-level OOD Detection Methods* | | | | | |
| **Ours** | VLM (10/16-shot) | 74.6 | 4.5 | 38.2 | 3.0 |

Table 11. Average AUROC (↑) scores comparison between our proposed method and oracle (upper bound via exhaustive searching) based on four test cases (BDD-100k→OpenImages, BDD-100k→MS-COCO, PASCAL-VOC→OpenImages, and PASCAL-VOC→MS-COCO) of object-level OOD detection tasks.

| **Configurations** | Oracle (upper bound) | **Ours** |
|---|---|---|
| Average AUROC | 98.49±0.16 | 96.44±0.18 |
| Average inference time (ms) | 5170.9 | 41.2 |

## D.8. Ablation Study on the Number of Fine-tuning Samples (shots)

**Object-level OOD detection.** We explore the impact of the number of fine-tuning samples (shots) for the object-level OOD detection task based on FPR95 scores in Figure 6. Note that the performance numbers for traditional object-level OOD detection methods are also provided in the figure, represented by dashed lines. However, they always require the *entire* ID dataset during the training procedure and thus could *not* be applied for the few-shot fine-tuning paradigm. Their numbers are *only* used as a performance reference.

We follow the setup in RUNA [55] and provide a performance comparison for 1-shot, 5-shot, and 10-shot cases in Figure 6. It is observed that our proposed method outperforms all existing traditional object-level OOD detection

methods across all test cases, requiring orders-of-magnitude fewer training samples. This is achieved by leveraging the extensive knowledge from both the vision and text domains during the pre-training stage for VLMs, such as CLIP. In addition, the increase in the number of fine-tuning samples also significantly further boosts the performance of our proposed method across all test cases. Moreover, our proposed method consistently outperforms RUNA in every fine-tuning number ablation setup across all test cases in object-level OOD detection tasks, further demonstrating the effectiveness of our method in capturing better image representations for both the local target object and the global background scene in object-level OOD detection tasks.

**Image-level OOD detection.** In Table 2, we show that our proposed method is the new SOTA method for the image-level OOD detection task based on the most popular 16-shot fine-tuning setup. In this section, we further evaluate our method with an ablation study on the number of fine-tuning samples for image-level OOD detection tasks, as shown in Tables 12 and 13. In 1-shot fine-tuning, our proposed method again demonstrates strong OOD detection performance for image-level OOD detection tasks, securing the lead position across all test cases. Similarly, our proposed method also outperforms all competing methods in the 4-shot fine-tuning setup. This study further illustrates the benefits of our proposed method, which utilizes patch output embeddings in addition to the `[CLS]` output embedding and the text embedding that explains all objects in the background image, together with our proposed weights, which optimally combine those contextual representations together and thereby capture fine-grained visual details of the target object and relevant spurious cues in background for the visual domain.

Table 12. AUROC (↑) scores for 1-shot image-level OOD detection results with ImageNet-1k as ID dataset. **Bold** and underlined numbers represent best and second-best results, respectively.

| Method | OOD Dataset | | | |
|---|---|---|---|---|
| | iNaturalist | SUN | TEXTURE | PLACES |
| CoOp [59] | 94.82 | 90.85 | 86.23 | 88.18 |
| PLOT [2] | 95.33 | 91.54 | 88.91 | 89.75 |
| LoCoOp [33] | 96.06 | 94.27 | 89.39 | 91.18 |
| GalLoP [21] | 96.42 | 93.34 | 89.76 | 90.63 |
| DPM-T [56] | 97.81 | 96.07 | 93.96 | 90.81 |
| SCT [54] | 95.36 | 94.83 | 88.56 | 91.74 |
| ID-like [1] | 96.85 | 89.34 | 90.69 | 86.86 |
| NegPrompt [24] | 97.93 | 94.75 | 90.80 | 92.54 |
| **Ours** | **98.24**±0.09 | **96.31**±0.16 | **94.53**±0.27 | **92.89**±0.21 |

## D.9. Ablation Study on Different Backbones of CLIP Image Encoder

**Object-level OOD detection.** Following the setup in RUNA [55], we report the performance comparison of our method with other competing approaches for the object-level OOD detection task with ViT-B/16 backbone for the image encoder of CLIP in Table 1. In this section, we present a further ablation study to evaluate the impact of different backbone configurations of the CLIP image encoder on object-level OOD detection performance, as shown in Table 14. It reveals that larger backbone models yield better performance across all object-level OOD detection test cases. To be specific, a larger ViT model is equipped with a higher number of layers, attention heads per layer, and an increased embedding dimension, which helps it extract and capture more detailed visual representations and therefore contribute to better performance in OOD detection. Furthermore, another factor which is specifically important to our proposed framework is the number of patch output embeddings. This is because our proposed weights combine the `[CLS]` and patch output embeddings to help capture the fine-grained visual details of the target object and relevant spurious cues in the background. For example, the smallest model (ViT-B/32) produces only 49 image patches for the entire input image (both target object and background image), while the largest one utilizes 576 image patches for the same input image. This results in a significant difference in the granularity of the vision domain information that is captured by the CLIP image encoder and utilized by our proposed method. A higher number of image patches for the same input image would contribute to more fine-grained details, and, when used adequately with weight enhancement in our proposed method, becomes significantly more important for tasks like object-level OOD detection.

Table 13. AUROC (↑) scores for 4-shot image-level OOD detection results with ImageNet-1k as ID dataset. **Bold** and underlined numbers represent best and second-best results, respectively.

| Method | OOD Dataset | | | |
|---|---|---|---|---|
| | iNaturalist | SUN | TEXTURE | PLACES |
| CoOp [59] | 95.72 | 91.79 | 87.24 | 89.11 |
| PLOT [2] | 96.03 | 92.21 | 89.60 | 90.45 |
| LoCoOp [33] | 96.43 | 94.71 | 89.86 | 91.60 |
| GalLoP [21] | 96.89 | 93.78 | 90.12 | 91.04 |
| DPM-T [56] | 98.65 | 96.50 | 94.48 | 91.50 |
| SCT [54] | 95.68 | 95.07 | 88.91 | 92.02 |
| ID-like [1] | 97.52 | 89.89 | 91.31 | 87.57 |
| NegPrompt [24] | 98.51 | 95.20 | 91.42 | 93.01 |
| **Ours** | **98.81**±0.11 | **96.90**±0.14 | **95.38**±0.22 | **93.66**±0.25 |

**Image-level OOD detection.** We also conduct the ablation study on the backbone configuration of the CLIP image encoder for image-level OOD detection tasks, as shown in Table 15. Similar to observations in object-level OOD detection tasks, it reveals that the number of image patches also impacts image-level OOD detection performance. Additionally, it is observed that the popular configuration of

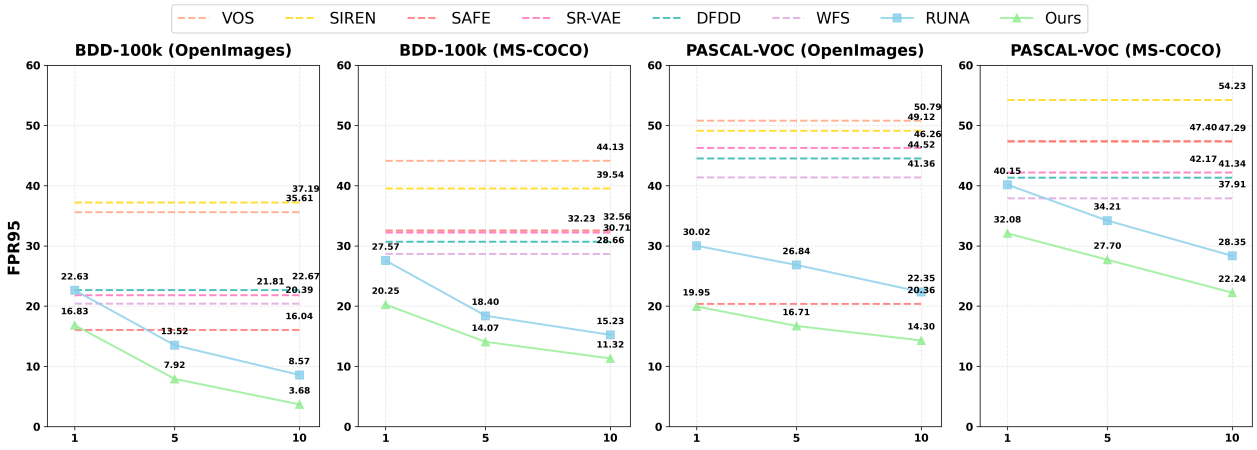


Figure 6. Ablation study on the number of fine-tuning samples (shots) based on FPR95 (↓) scores for object-level OOD detection tasks. "X (Y)" denotes test case with "X" as ID dataset and "Y" as OOD dataset. The dashed lines indicate reference numbers of traditional object-level OOD detection methods with the *entire* ID dataset for training, as it could *not* be applied for few-shot fine-tuning.

Table 14. Results of ablation study on object-level OOD detection tasks for the backbone configuration of CLIP image encoder. "# of patches" indicates the number of patch output embeddings that the CLIP image encoder will produce.

| Backbone | # of patches | BDD-100k | | | | PASCAL-VOC | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | OpenImages | | MS-COCO | | OpenImages | | MS-COCO | |
| | | AUROC ↑ | FPR95 ↓ | AUROC ↑ | FPR95 ↓ | AUROC ↑ | FPR95 ↓ | AUROC ↑ | FPR95 ↓ |
| ViT-B/32 | 49 | 97.65 | 6.23 | 95.10 | 13.08 | 95.37 | 17.92 | 94.11 | 24.72 |
| ViT-B/16 | 196 | 98.52 | 3.68 | 95.91 | 11.32 | 96.25 | 14.30 | 95.07 | 22.24 |
| ViT-L/14 | 256 | 98.73 | 3.20 | 96.05 | 10.27 | 96.42 | 14.01 | 95.35 | 21.98 |
| ViT-L/14@336px | 576 | 99.24 | 2.72 | 96.39 | 10.08 | 96.88 | 13.37 | 95.85 | 20.65 |

ViT-B/16 already delivers excellent image-level OOD detection performance, and a larger model like ViT-L/14@336px would likely contribute only a modest performance improvement at a potentially higher computational cost (e.g., ViT-L/14@336px is equipped with 24 layers, 1024 width, and 16 attention heads per layer). This suggests that ViT-B/16 would be the optimal configuration for the CLIP image encoder, striking a balance between efficiency and performance for our framework on image-level OOD detection tasks.

## D.10. Discussions of Component Designs

In this subsection, we present extended discussions of the component design of our proposed method, including the choice of shared weights for encoders, the design choice against RUNA, and the use of CLIP's projection matrix for patch tokens.

**Shared weights for encoders.** A natural question regarding the shared weights design for encoders in our proposed method is whether our cross-encoder attention mechanisms (i.e., *inter*-image attention (Eq. (5)) and *inter*-text attention (Eq. (7))) remain well-defined when queries and keys originate from different encoder instances. We emphasize that both pairs of encoders (Image Encoders 1 and 2, Text Encoders 1 and 2) are initialized from the same pretrained CLIP checkpoint and remain *frozen* throughout training; only the lightweight CNN combiners and MLP layers are updated. Consequently, the query and key projection matrices $W_Q^{\text{img},l,h}$ and $W_K^{\text{img},l,h}$ in Image Encoders 1 and 2 are *numerically identical* at every layer and head, and likewise for the text encoder projections $W_Q^{\text{text},l,h}$ and $W_K^{\text{text},l,h}$. This guarantees that queries and keys reside in precisely the same representational subspace, making their inner products semantically meaningful. In *inter*-image attention, the `[CLS]` query from Image Encoder 1 attends to patch keys from Image Encoder 2 using the same $W_Q$ and $W_K$ that were jointly optimized during CLIP's pretraining to capture visual relationships. The only difference between the two encoder instances is the *input*: Image Encoder 1 receives the cropped target object $I_x$, while Image Encoder 2 receives the masked background $I_{(x)}$. The cross-encoder attention, therefore, measures how strongly the holistic representation of the target object relates to each spatial location in the background, which is a semantically well-grounded operation within a shared embedding space.

In addition, we present the results of an ablation study on shared weights design (i.e., the encoders (Image Encoders 1

Table 15. Results of ablation study on image-level OOD detection tasks for the backbone configuration of CLIP image encoder. "# of patches" indicates the number of patch output embeddings that the CLIP image encoder will produce.

| Backbone | # of patches | iNaturalist | | SUN | | TEXTURE | | PLACES | |
|---|---|---|---|---|---|---|---|---|---|
| | | AUROC ↑ | FPR95 ↓ | AUROC ↑ | FPR95 ↓ | AUROC ↑ | FPR95 ↓ | AUROC ↑ | FPR95 ↓ |
| ViT-B/32 | 49 | 98.72 | 6.34 | 96.81 | 17.15 | 95.48 | 18.46 | 93.34 | 26.90 |
| ViT-B/16 | 196 | 99.16 | 4.92 | 97.35 | 16.40 | 96.52 | 16.91 | 94.27 | 25.47 |
| ViT-L/14 | 256 | 99.28 | 4.75 | 97.67 | 16.03 | 96.76 | 16.50 | 94.63 | 25.11 |
| ViT-L/14@336px | 576 | 99.64 | 4.20 | 98.00 | 14.81 | 97.02 | 15.65 | 94.98 | 24.32 |

and 2, Text Encoders 1 and 2) are still initialized from the same pretrained CLIP checkpoint but kept as independent instances) for encoders in Table 16. In our proposed method, the object and background encoders share weights, which is able to achieve equivalent performance to separate weights while *halving* memory usage.

Table 16. Results of ablation study on object-level OOD detection tasks for the shared weights and project representations configurations of CLIP encoders. Numbers reported are average AUROC (↑) scores based on four test cases (BDD-100k→OpenImages, BDD-100k→MS-COCO, PASCAL-VOC→OpenImages, and PASCAL-VOC→MS-COCO) of object-level OOD detection tasks.

| Configurations | Average AUROC ↑ |
|---|---|
| W/o shared weights | 96.42±0.17 |
| Project global representation only | 95.07±0.42 |
| Project patch tokens only | 95.51±0.49 |
| **Ours (w/ shared weights; project both)** | 96.44±0.18 |

**Component designs against RUNA.** Table 17 decomposes the performance gains on the image-level OOD detection tasks of our proposed method over RUNA. It shows that patch-level features for the target object contribute most substantially, as they capture fine-grained visual details that RUNA discards. The next most significant contributor is background modeling.

Table 17. Results of ablation study on image-level OOD detection tasks for component designs against RUNA. Numbers reported are average AUROC (↑) scores based on four test cases of image-level OOD detection tasks with ImageNet-1k as ID dataset.

| Configurations | Average AUROC ↑ |
|---|---|
| W/o patch-level features for the target object | 94.64±0.30 |
| W/o background modeling | 94.96±0.31 |
| W/o unified inference | 96.47±0.17 |
| **Ours (Full)** | 96.83±0.15 |

**Applying CLIP's projection matrix to patch tokens.** In the standard CLIP pipeline, the projection matrix $P \in \mathbb{R}^{\bar{d} \times d^{\text{img}}}$ is applied exclusively to the `[CLS]` embedding $z_{0,x}^{\text{img},L}$ to map it into the joint vision–language space of dimension $\bar{d}$. In our framework, we extend the application of $P$ to all patch embeddings $\{z_{i,x}^{\text{img},L}\}_{i=1}^{N}$ (Eq. (3)), which raises the question of whether this extension is semantically justified, given that $P$ was trained only on `[CLS]` representations.

We argue that this extension is both mathematically sound and empirically validated. First, the `[CLS]` token and all patch tokens share the same embedding space $\mathbb{R}^{d^{\text{img}}}$ at the output of the final ViT layer $L$. After $L$ layers of multi-head self-attention, every patch token has been contextually enriched through extensive interaction with all other tokens, including the `[CLS]` token. As a result, individual patch embeddings at layer $L$ are not raw local features: they are globally informed representations that encode both local spatial detail and broader contextual information. Since $P$ is a linear projection from $\mathbb{R}^{d^{\text{img}}}$ to $\mathbb{R}^{\bar{d}}$, it is applicable to any vector in $\mathbb{R}^{d^{\text{img}}}$, including patch tokens, and its effect is to project these representations into the vision–language alignment space. This observation is also consistent with recent findings in dense vision–language modeling [38, 39, 58], which demonstrate that CLIP's patch-level features, when projected into the joint embedding space, yield spatially localized representations that are highly aligned with textual semantics and suitable for dense prediction tasks such as segmentation and grounding.

Furthermore, the practical motivation for projecting patch tokens with $P$ is to enable our text–vision alignment weights $\{\mu_{i,x}\}_{i=0}^{N}$ (Eq. (3)), which measure the cosine similarity between each projected patch embedding $Pz_{i,x}^{\text{img},L}$ and the text embedding $M_{\text{text}}(t_x)$. Without this projection, patch tokens would remain in the visual-only space $\mathbb{R}^{d^{\text{img}}}$, where direct comparison with text embeddings in $\mathbb{R}^{\bar{d}}$ is not possible. By applying $P$, we bring each patch into the shared vision–language space, thereby enabling a spatially fine-grained assessment of which patches are most semantically consistent with the textual description of the target object. Also, our ablation studies (Tables 3 and 4) provide direct empirical evidence: removing the text–vision alignment module, which depends on applying $P$ to patch tokens, consistently degrades performance across all object- and image-level OOD detection benchmarks. This confirms that the projected patch tokens carry meaningful, complementary information that strengthens OOD detection beyond what the `[CLS]`-only pipeline can achieve.