Any Detector Can Detect Anything
Supplementary Material

Thomas E. Huang[1*] Siyuan Li[1*] Martin Danelljan[1] Henghui Ding[1]
Luc Van Gool[1,2] Fisher Yu[1]
[1] Computer Vision Lab, ETH Zürich [2] INSAIT, Sofia University “St. Kliment Ohridski”

The supplementary material is structured as follows:

- Section A: Additional discussion.
- Section B: Additional ablation studies and analysis.
- Section C: Additional results with different backbones.
- Section D: Comprehensive qualitative analysis of the results, highlighting the robust detection performance achieved by ADDA.
- Section E: Details about our model architecture and the training procedure.

A. Discussion

In this section, we provide additional discussion about details that could not fit in the main manuscript.

A.1. One-Shot vs. Few-Shot Object Detection

The domains of One-Shot Object Detection (OSOD) and Few-Shot Object Detection (FSOD) are closely intertwined yet distinct, with key differences that set them apart as fundamentally different tasks. Understanding these differences is crucial before performing any direct comparison of methods designed for each task. In this section, we outline the primary distinctions between OSOD and FSOD.

1. **Template Variability**: OSOD strictly focuses on scenarios with only one visual template, whereas FSOD allows for the use of multiple templates. This distinction implies that most methods developed for OSOD cannot be directly applied to FSOD, as they lack the capability to handle multiple visual templates.
2. **Primary Focus**: The primary goal of OSOD is to generalize to novel (*i.e*., unseen) classes without any prior exposure. In contrast, FSOD aims to maximize performance on “novel” classes, often employing novel fine-tuning techniques. Therefore, while OSOD methods operate without access to data from novel classes, FSOD methods frequently utilize such data for fine-tuning to rapidly adapt to new classes with limited examples.
3. **Assumptions about Novel Classes**: OSOD operates without any assumptions regarding novel classes and treats them independently. Novel classes are processed one by one, and common post-processing techniques like inter-class non-maximum suppression (NMS) are not employed in OSOD, as they rely on knowledge of other novel classes. This is done to emulate real-world scenarios where assumptions about test-time conditions cannot be made. In contrast, FSOD does not impose such constraints as they focus on adaptation.

The ability of ADDA to perform effectively in both OSOD and FSOD benchmarks is a testament to its versatility and robustness. This adaptability indicates that our method is not only capable of handling the strict constraints of OSOD but also excels when presented with multiple visual templates in FSOD. The successful application across these different settings underscores the superior performance and generalization capabilities of our approach.

A.2. Different Detection Architectures

Although ADDA can work with any detector to perform visual prompt-based detection, different types of detectors have their strengths and weaknesses when applied to this task. We provide some discussion and intuition for when and where each type of detector is the most effective based on our empirical results.

Overall, we found one-stage detectors to be the strongest for our task and achieves the best balance between generalization and efficiency. Across multiple benchmarks, FCOS [11] and GFL [6] achieve the best results on novel classes. Two-stage detectors, on the other hand, performs worse, especially on base classes. This makes sense, as our framework only does binary classification within the detection head, so the complex design of two-stage detectors complicates training and limits performance.

Recently, query-based detectors [1, 15] have become a popular approach for object detection. Query-based detectors utilize a set of learnable queries and maps them to corresponding object bounding boxes. We found query-based detectors to be very strong at detecting objects they are trained on, demonstrated by their superior performance on base classes. However, when generalizing the learned queries to

[*] Equal contribution.

Table 1. Sensitivity analysis on the R-OSOD COCO benchmark using the COCO 2017 validation set across multiple random seeds.

Method	Backbone	Base $\text{AP}^{5\text{N}}_{50}$					Novel $\text{AP}^{5\text{N}}_{50}$				
		S1	S2	S3	S4	Avg.	S1	S2	S3	S4	Avg.
BHRL [14]	ResNet-50	45.5±0.35	39.9±0.37	40.0±0.43	41.1±0.53	41.6±0.42	13.9±0.71	17.3±0.57	11.1±0.47	14.0±0.61	14.1±0.59
ADDA	CvT-13	**51.7**±0.27	**47.9**±0.33	**48.6**±0.37	**49.5**±0.31	**49.6**±0.32	**17.7**±0.76	**18.7**±0.73	**16.2**±0.65	**17.3**±0.69	**17.5**±0.71

Table 2. Ablation study of Bidirectional Feature Passing (BFP) iterations on *split 1* of the R-OSOD COCO benchmark.

BFP Iterations	Base $\text{AP}^{5\text{N}}_{50}$	Novel $\text{AP}^{5\text{N}}_{50}$	FLOPs (G)
0	50.1	17.2	-
2	51.4	17.4	3.2
4	52.3	17.4	6.4
6	52.5	**17.7**	9.6
8	**52.7**	17.4	12.8

novel classes, their performance becomes much weaker, indicating that queries can more easily overfit to the base classes and lead to worse generalization. We leave improving this aspect of query-based detectors as future work.

B. Additional Ablation Studies and Analysis

In this section, we provide additional ablation studies and analysis to validate and understand our framework.

B.1. Sensitivity Analysis

In this section, we present a sensitivity analysis of model performance with respect to different visual templates. The results are shown in Table 1. Here, we use 20 different seeds for both BHRL [14] and ADDA, utilizing varying visual templates for each object class in each seed. Subsequently, we compute the 95% confidence interval of the $\text{AP}^{5\text{N}}_{50}$ scores for each model.

The analysis reveals that the performance of both models remains stable across varying seeds, demonstrating less than a 1 point variation in $\text{AP}^{5\text{N}}_{50}$ for both base and novel classes. Considering the notable AP gains of ADDA compared to BHRL, coupled with the low variance observed across the 20 seeds, it clearly demonstrates the superior performance of our proposed model.

B.2. Bidirectional Feature Passing Iterations

We conduct an ablation study on the effect of the number of iterations of our Bidirectional Feature Passing (BFP) module in Table 2. With only two iterations, BFP can improve the performance on base classes by 1.3 points and on novel classes by 0.2 points, with a minor increase in computation. At six iterations, BFP achieves the best performance of 17.7 points on novel $\text{AP}^{5\text{N}}_{50}$. With even more iterations, performance on the base classes can be even better, but accuracy on novel classes drops. We use six iterations in all other experiments for the best balance between performance and computational efficiency, unless otherwise specified.

B.3. Attention Visualizations

We visualize attention operations within the Template-Aware Backbone for the final three blocks and different inputs in Figure 1. The attention operations include both self-attention of target image and cross-attention between target image and visual template. For each target image, we show attention operations for both a positive template and a negative template. With positive templates, our self-attention attends to the relevant objects in the target image (row 1), while cross-attention focuses on the object region in the template (row 2). For negative templates, both attentions are more scattered across the image (rows 3 and 4).

C. Additional Backbones

We investigate how our framework performs with stronger backbones and weight initialization. In addition to CvT [13] and Swin Transformer [8], we use the recently developed DINOv2 [9] models that are built on the Vision Transformer (ViT) [3] architecture. We also investigate stronger pre-training weights on the ImageNet-22K [2] dataset. For these models, we adhere to the same training protocol as detailed in section E.2, but with a reduced training schedule by 25% (9 training epochs). This adjustment is necessary as we observe a tendency for these models to overfit to the base data, leading to decreased performance on novel classes. The results are presented in Table 3.

Our analysis reveals some interesting insights. The Swin Transformer backbone, with its window-based attention mechanism, shows enhanced efficiency in modeling feature interactions at higher resolutions. However, its limited cross-window feature propagation in the final blocks tends to favour base class performance, resulting in weaker generalization to novel classes. Improvements can be achieved with ImageNet-22K pre-training weights, which offer a boost of +2.9 in base $\text{AP}^{5\text{N}}_{50}$ and +0.9 in novel $\text{AP}^{5\text{N}}_{50}$. A larger backbone, Swin-S, further improves performance, adding +6.2 in base $\text{AP}^{5\text{N}}_{50}$ and +0.3 in novel $\text{AP}^{5\text{N}}_{50}$. DINOv2 focuses on learning general-purpose visual features by applying self-supervised learning on a large curated dataset, LVD-142M [9]. We adopt their pre-trained weights as initialization. The results show that DINOv2 provides very strong visual features as advertised, achieving the best base $\text{AP}^{5\text{N}}_{50}$ of 69.9 and a novel $\text{AP}^{5\text{N}}_{50}$ of 15.0.

Despite these encouraging results, especially in base class detection, the overall performance in novel classes remains a challenge. This highlights the inherent difficulty in achiev-

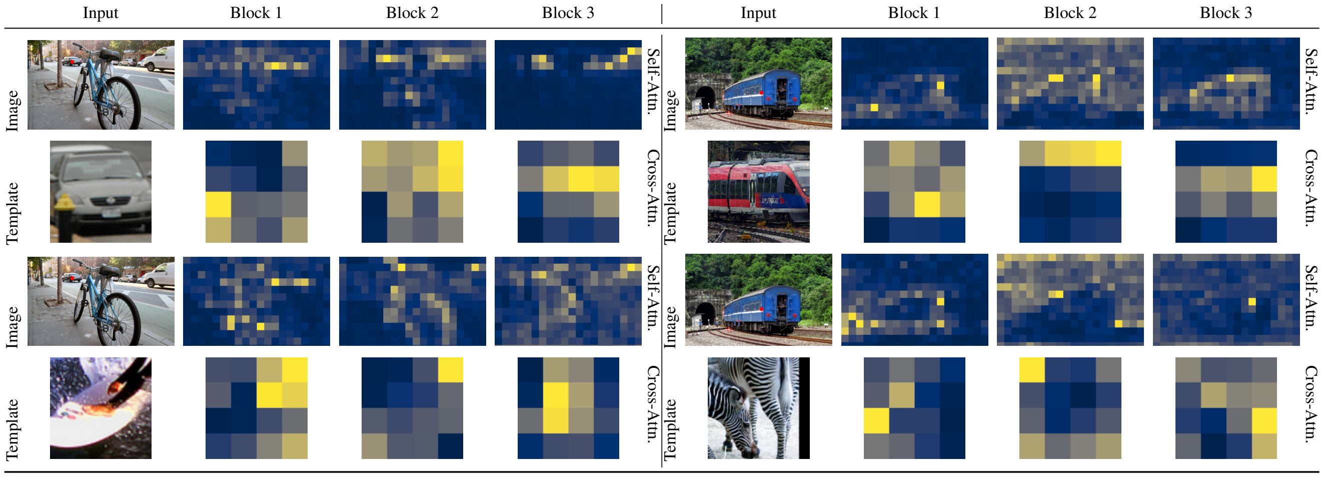


Figure 1. Visualization of attention operations in the Template-Aware Backbone for the final three blocks, including target image self-attention and image-template cross-attention. The red point indicates the location of the query point for attention and yellowness indicates higher attention values. Best viewed in color.

Table 3. Different weight initialization and backbone networks for ADDA using FCOS on the R-OSOD COCO benchmark using the COCO 2017 validation set. IN stands for ImageNet pre-training.

Weight Initialization	Backbone	# Backbone Params. (M)	Base AP^{5N}_{50}					Novel AP^{5N}_{50}				
			S1	S2	S3	S4	*Avg.*	S1	S2	S3	S4	*Avg.*
IN-1K	Swin-T [8]	28.0	55.5	51.9	52.0	53.1	53.1	13.5	14.6	11.6	11.9	12.9
	CvT-13 [13]	20.0	52.5	47.9	48.5	49.6	49.6	17.7	**18.8**	**16.3**	**17.1**	**17.5**
IN-22K	Swin-T [8]	28.0	58.4	54.4	55.4	55.9	56.0	14.8	14.7	12.2	13.7	13.8
	Swin-S [8]	50.0	63.4	61.1	62.0	62.4	62.2	16.6	15.2	11.6	13.0	14.1
LVD-142M [9]	DINOv2-B [9]	86.0	**70.3**	**68.8**	**70.8**	**69.6**	**69.9**	**18.0**	14.3	15.5	12.3	15.0

ing generalization in visual prompt-based detection tasks, predominantly due to overfitting issues. We hope our findings will stimulate further research in this area, aiming to bridge the gap in performance between base and novel class detection.

D. Additional Qualitative Results

In this section, we provide additional qualitative results of our detector ADDA. We first compare results with BHRL [14] on split 3 of COCO [7] for positive templates in Figure 2 and for negative templates in Figure 3. With positive templates, ADDA can better localize objects of interest and produces fewer false positives. With negative templates, ADDA do not incorrectly output as many detections across both base and novel classes.

Furthermore, we provide a way to perform visual template-based detection based on text inputs, and the results are shown in Figure 4. To do this, we first utilize Stable Diffusion [10] to generate visual templates based on the textual description. This enables the user to fine-tune the generated templates visually to better match the target of interest. Then, we input the generated templates to ADDA to detect the corresponding objects in each target image. The results show that ADDA supports diffusion-generated templates without requiring re-training, further demonstrating the generalizability of our method.

Finally, we also show many more qualitative results on base classes in Figure 5 and on novel classes in Figure 6. We also provide failure cases of ADDA in Figure 7, which are mostly caused by ambiguous and low-quality visual templates.

E. Additional Details

We provide additional details regarding the architecture, the training procedure, and our R-OSOD benchmarks.

E.1. Architecture Details

ADDA is composed of a Template-Aware Adapter and a standard detection head. Template-Aware Adapter is further decomposed into Template-Aware Backbone, Template-Aware Pyramid, and Bidirectional Feature Passing.

E.1.1. Template-Aware Backbone

In this work, we use three different Transformer base models, Vision Transformer (ViT) [3], CvT [13], and Swin Transformer [8]. We use these backbones to demonstrate that our design can generalize to different types of Transformer base models.

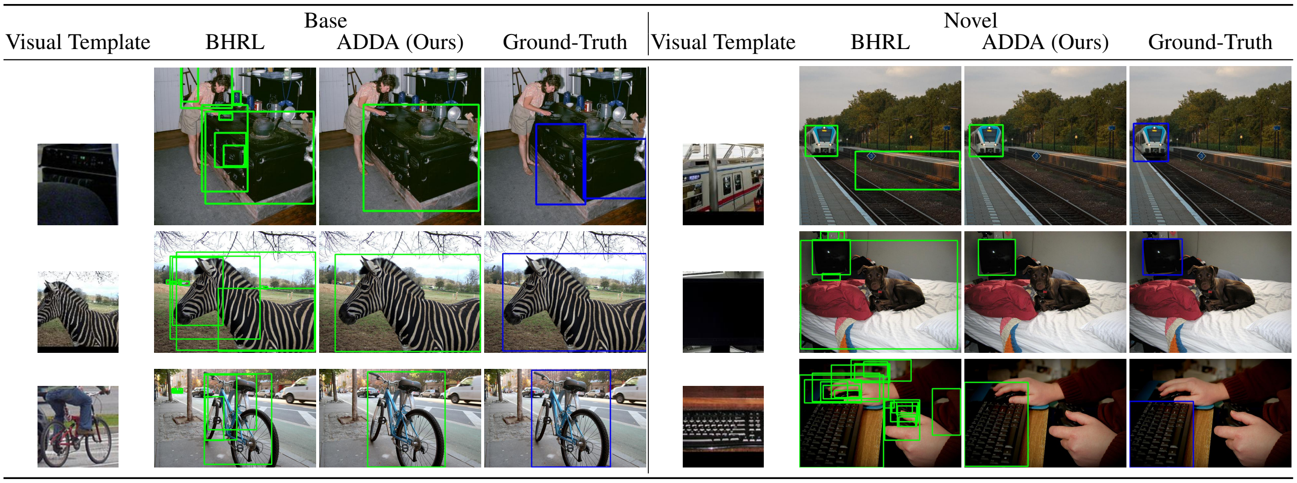


Figure 2. Additional qualitative comparison between BHRL [14] and ADDA with **positive** visual templates on COCO [7]. Our detector produces more accurate detections and fewer false positives.

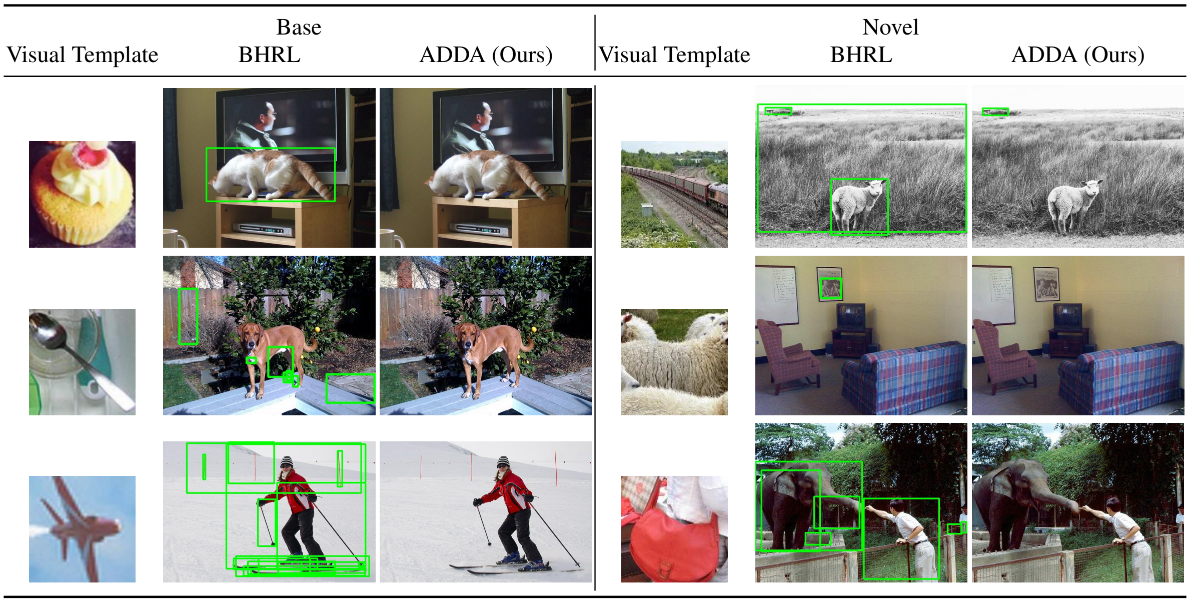


Figure 3. Additional qualitative comparison between BHRL [14] and ADDA with **negative** visual templates on COCO [7]. Our detector can better handle negative templates and produces fewer false positive detections.

ViT. ViT consists of 12 blocks, which we split into four stages with 3 blocks each. For our Template-Aware Backbone, we replace the final 4 blocks with FEIBs. The attention operation used in ViT is the standard Multi-Head Attention [12] operation, and we include an additional image-template cross-attention operation in each FEIB. The backbone yields a feature map after stages 2, 3, and 4 with a feature dimension of 768.

CvT. CvT-13 consists of three stages with 1, 2, and 10 blocks, respectively. For our Template-Aware Backbone, we replace the final 4 blocks with FEIBs. The image-template cross-attention module in each FEIB contains three convolutional projection layers which yield queries, keys, and values for the attention operation. The query projection layer uses 3×3 kernel size with stride 1 and padding 1, while the key and value projection layers use a stride of 2 to reduce the dimensions. The backbone yields one set of feature maps after each stage. The feature dimension for each feature map is 64, 192, and 384, respectively.

Swin Transformer. Swin-T consists of four stages with 2, 2, 6, and 2 blocks, respectively. Swin-S instead contains 18 blocks in the third stage. For our Template-Aware Backbone, we replace the final 2 blocks of stage 3 and all 2 blocks of stage 4 with FEIBs, for a total of 4 blocks. For the image-template cross-attention module in each FEIB, we perform window-based cross-attention by attending tokens

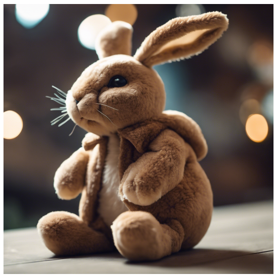
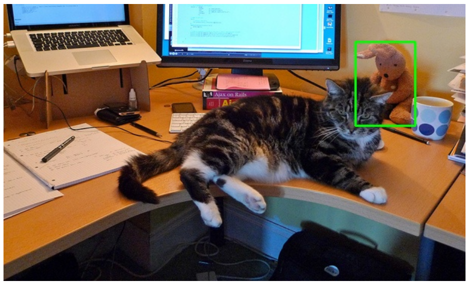
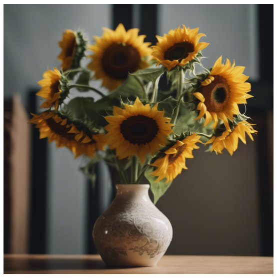
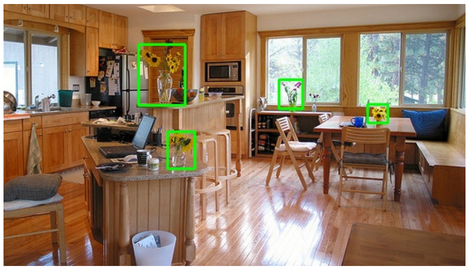
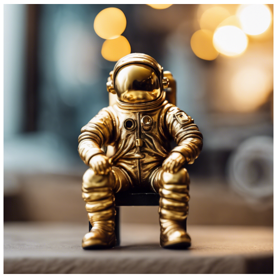
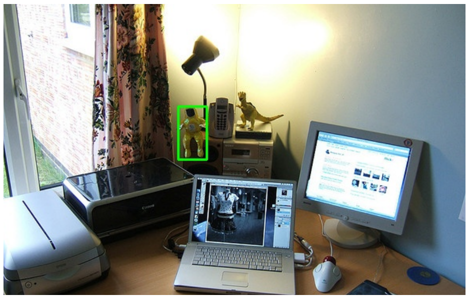
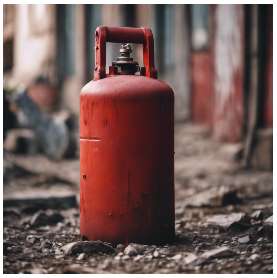
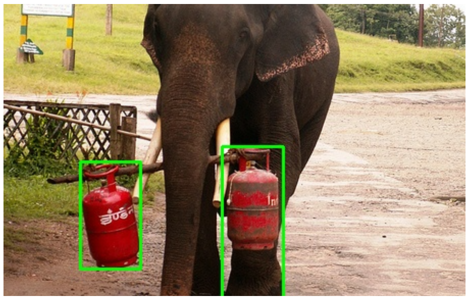
Text Prompt	Visual Template	ADDA Predictions
Light brown colored stuffed rabbit, sitting down and slouched forward		
Multiple sunflowers in a single vase		
Toy gold colored astronaut in a sitting position		
Red colored gas cylinder, in a rough condition		

Figure 4. Qualitative results of ADDA using visual templates generated with Stable Diffusion [10] using the text prompts. ADDA supports diffusion-generated templates without requiring re-training.

in each window to all the template tokens. In implementation, this is equivalent to duplicating the template tokens for all windows and then computing attention on the concatenation of window and template tokens. The backbone yields a feature map after stages 2, 3, and 4 with feature dimensions 192, 384, and 768, respectively.

E.1.2. Template-Aware Pyramid

Template-Aware Pyramid (TAP) contains four FEIBs for integrating template features during the construction of the feature pyramid. The FPN takes as input three image feature maps F_t^1, F_t^2, F_t^3 with strides $2^3, 2^4$, and 2^5 and the final template feature map F_q from the backbone. We replace the existing 3×3 convolutional layers in a standard FPN with a FEIB each. The first three FEIBs are applied in a top-down manner starting from the third level, and we use a convolutional tokenizer with kernel size 3×3, stride 1, and padding 1. The last FEIB is applied on top of the third level to additionally downsample the feature map to half the spatial dimensions, and thus we use the same convolutional tokenizer but with stride 2.

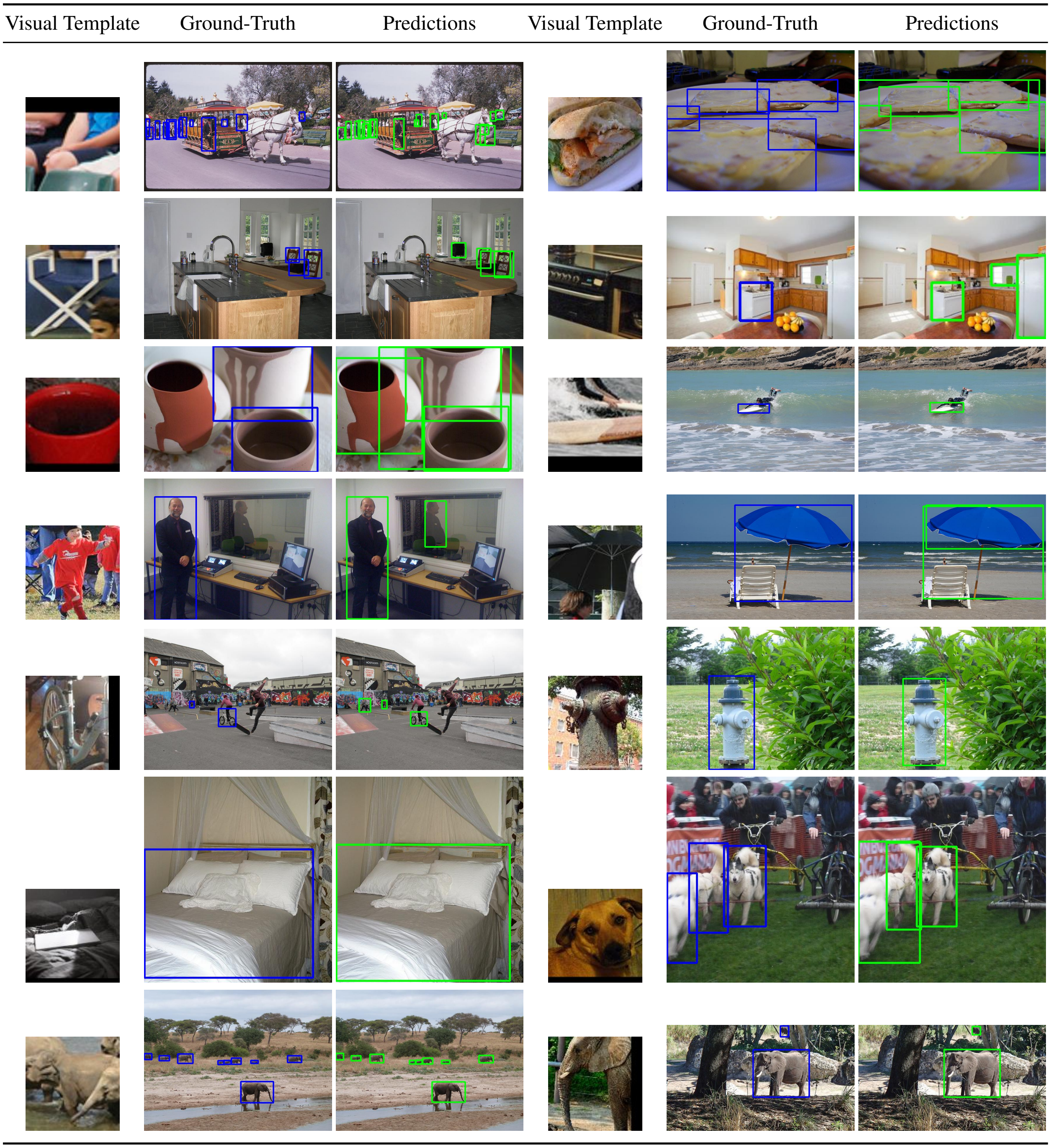


Figure 5. Additional qualitative results of ADDA on **base** classes of split 3 of COCO [7].

E.1.3. Bidirectional Feature Passing

For our Bidirectional Feature Passing (BFP) module, we use a shared convolutional layer across all feature levels with kernel size 3 × 3 and padding 1 to generate the dynamic offset and modulation factors for each spatial location.

E.2. Training Details

For target images, we resize them to a fixed input resolution (detailed below) while maintaining the aspect ratio. For visual templates, we take a center square crop of the given bounding boxes and resize to the input resolution of 128 × 128. During training, we randomly horizontally flip images and templates with a probability of 0.5. For OSOD, we randomly sample one visual template for each image in the

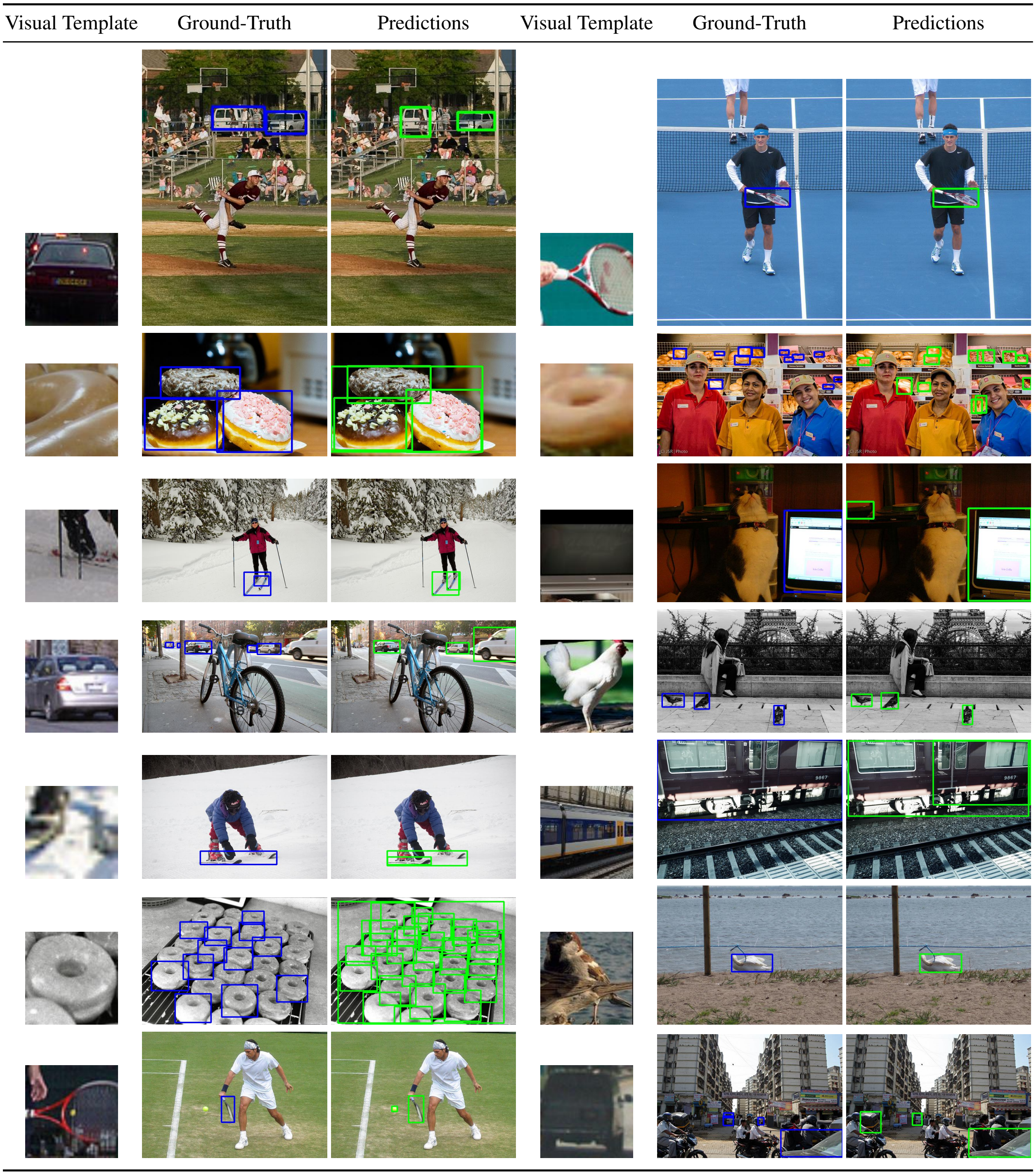


Figure 6. Additional qualitative results of ADDA on **novel** classes of split 3 of COCO [7].

batch to be consistent with prior methods [14]. For FSOD, we randomly sample between one and ten visual templates to enable generalization to multiple templates. By default, we use learning rate warm-up with 500 steps at the start of training. All models are trained on 8 GeForce RTX 3090 GPUs.

ViT / CvT. We use an input resolution of 640×640, as the dense attention operations in ViT and CvT are too costly for higher resolution inputs.

Swin. We use an input resolution of 1024×1024 due to the computational efficiency of window-based attention used in Swin Transformer.

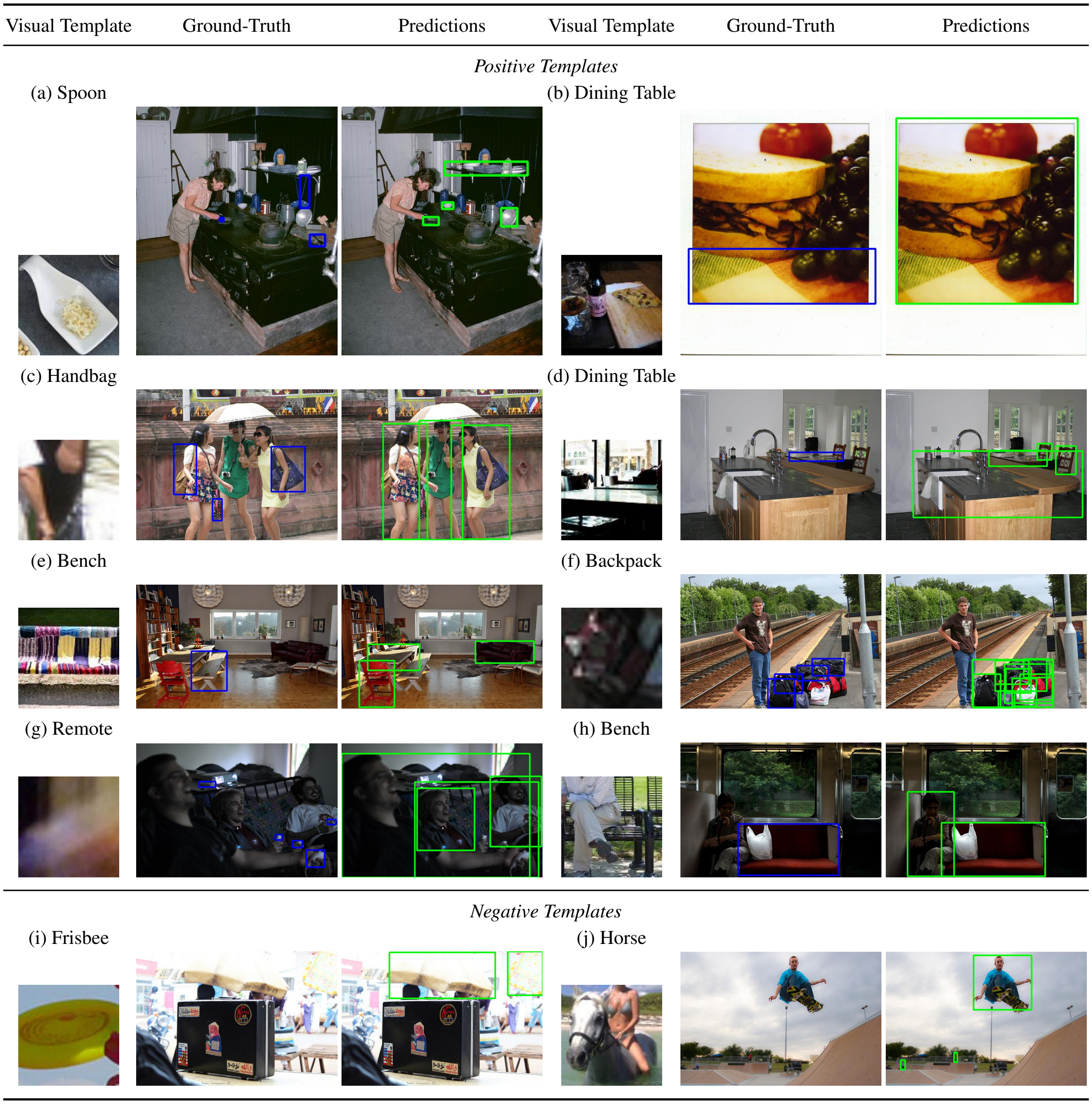


Figure 7. Failure cases of ADDA on split 3 of COCO [7] with both positive and negative visual templates. A majority of the issues are due to ambiguous visual templates, such as objects that look like another class (*e.g*., (a), (e), (f), (i)) and templates that include multiple objects (*e.g*., (b), (d), (h), (j)). Templates that include hand-held objects often contain hands and arms, which confuses the detector to output person detections (*e.g*., (c), (h), (j)). The quality of some visual templates is also too low (*e.g*., (g)).

E.3. R-OSOD Benchmarks Details

In this section, we provide additional details regarding our R-OSOD benchmarks.

LVISv1. LVISv1 [4] is annotated with three types of classes based on their frequency: frequent, common, and rare. In total, there are over 1200 classes. To evaluate generalization of one-shot methods, we use rare classes as the novel classes, which appear in only one to ten images. Frequent and common classes are used as base classes, which appear in more than ten images.

OpenImages. For OpenImages [5], we construct a zero-shot evaluation benchmark where we evaluate COCO-trained models directly on OpenImages without re-training. OpenImages contains 600 object classes, but some overlap with COCO classes. Furthermore, OpenImages's classes have a hierarchy, meaning some are parent classes and other are children classes. We remove the overlapped classes by filtering the OpenImages based on class name as well as the class hierarchy to ensure that all similar classes are removed. We also filter classes that do not have a sufficient number of images to use as visual templates during evaluation. After filtering, we have over 300 classes that we use as novel classes.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 1

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 2

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3

[4] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019. 8

[5] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020. 8

[6] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Advances in Neural Information Processing Systems*, 33:21002–21012, 2020. 1

[7] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 3, 4, 6, 7, 8

[8] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 2, 3

[9] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 3

[10] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 3, 5

[11] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019. 1

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4

[13] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22–31, 2021. 2, 3

[14] Hanqing Yang, Sijia Cai, Hualian Sheng, Bing Deng, Jianqiang Huang, Xiansheng Hua, Yong Tang, and Yu Zhang. Balanced and hierarchical relation learning for one-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 4, 7

[15] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 1