Supplementary Material of X-DETR: A Versatile Architecture for Instance-wise Vision-Language Tasks

Zhaowei Cai, Gukyeong Kwon, Avinash Ravichandran, Erhan Bas,
Zhuowen Tu, Rahul Bhotika, and Stefano Soatto

AWS AI Labs
{zhaoweic,gukyeong,ravinash,erhanbas,ztu,bhotikar,soattos}@amazon.com

In the supplementary, we provide more details and visualization examples which are not covered in the main paper.

A Losses

For completeness, we include the details of the losses we use in the paper.

A.1 Object Detection Losses

In total, there are three losses for detection: a binary cross-entropy loss $\mathcal{L}_{cls}$ for classification, a generalized IoU $\mathcal{L}_{iou}$ and L1 loss $\mathcal{L}_{L1}$ for bounding box regression,

$$\mathcal{L}_{det} = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{iou}\mathcal{L}_{iou} + \lambda_{L1}\mathcal{L}_{L1}, \tag{1}$$

where λ_{cls}, λ_{iou} and λ_{L1} are corresponding loss weights.

Classification The classification loss $\mathcal{L}_{cls}$ is a standard binary cross-entropy loss $\mathcal{L}_{ce}(y, \hat{y})$, where $y \in \{0, 1\}$ is the ground truth label and $\hat{y}$ is the prediction.

Bounding box regression Bounding box regression tries to regress to the target bounding box $\mathbf{b} = (b_{x_1}, b_{y_1}, b_{x_2}, b_{y_2})$ from a bounding box prediction $\hat{\mathbf{b}}$. $\mathcal{L}_{L1}$ is the L_1 loss,

$$\mathcal{L}_{L1}(\mathbf{b}, \hat{\mathbf{b}}) = ||\mathbf{b} - \hat{\mathbf{b}}||_1. \tag{2}$$

$\mathcal{L}_{iou}$ is the generalized IoU loss [11],

$$\mathcal{L}_{iou}(\mathbf{b}, \hat{\mathbf{b}}) = 1 - \Big(\frac{|\mathbf{b} \cap \hat{\mathbf{b}}|}{|\mathbf{b} \cup \hat{\mathbf{b}}|} - \frac{|\mathbf{b}_c \setminus \mathbf{b} \cup \hat{\mathbf{b}}|}{|\mathbf{b}_c|}\Big), \tag{3}$$

where $|\cdot|$ means the area of shape, and $\mathbf{b}_c$ is the smallest convex shape enclosing both $\mathbf{b}$ and $\hat{\mathbf{b}}$. More details can be found in [1, 11].

A.2 Vision-Language Alignment Losses

Object-Phrase Alignment Following [3], we used the contrastive loss of InfoNCE [8] to optimize for the object-phrase alignment. After the similarity matrix of every potential object-token pair is computed, the contrastive loss is applied for each row

(object-token alignment) $\mathcal{L}_{o2t}$ and each column (token-object alignment) $\mathcal{L}_{t2o}$ of this matrix. For each object,

$$\mathcal{L}_{o2t} = \frac{1}{|\mathcal{P}_o|} \sum_{j \in \mathcal{P}_o} -log \frac{\exp(o^T t_j/\tau)}{\sum_{k \in \mathcal{B}_t} \exp(o^T t_k/\tau)}, \tag{4}$$

where $\mathcal{P}_o$ is the positive text token set that object o should be aligned with, and $\mathcal{B}_t$ is the full text token set in this batch, t is the text token and τ is the temperature. For each text token,

$$\mathcal{L}_{t2o} = \frac{1}{|\mathcal{P}_t|} \sum_{j \in \mathcal{P}_t} -log \frac{\exp(t^T o_j/\tau)}{\sum_{k \in \mathcal{B}_o} \exp(t^T o_k/\tau)}, \tag{5}$$

where $\mathcal{P}_t$ is the positive object set that text token t should be aligned with, and $\mathcal{B}_o$ is the full object set in this batch.

Image-Caption Alignment Similar to CLIP, the loss is a cross-modality contrastive loss between the encoded image queries $\mathcal{B}_i = \{u\}$ and the captions $\mathcal{B}_c = \{v\}$. The image-caption contrastive loss is

$$\mathcal{L}_{i2c} = -log \frac{\exp(u^T v/\tau)}{\sum_{k \in \mathcal{B}_c} \exp(u^T v_k/\tau)}, \tag{6}$$

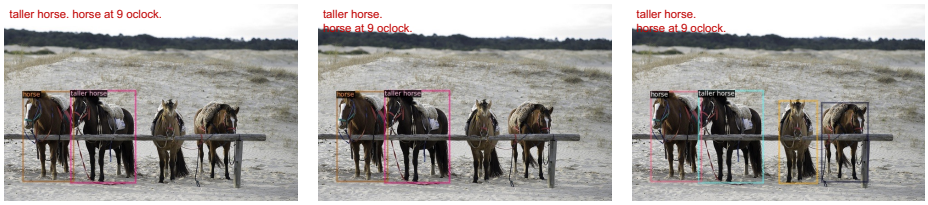
where u corresponds to v. And the caption-image contrastive loss is

$$\mathcal{L}_{c2i} = -log \frac{\exp(v^T u/\tau)}{\sum_{k \in \mathcal{B}_i} \exp(v^T u_k/\tau)}. \tag{7}$$

B Training Details

B.1 Pretraining

During pre-training on the joint datasets, X-DETR was trained for 10 epochs w.r.t. the *mixed* dataset. The batch size for fully/pseudo/weakly-annotated data is 4/2/4 for a single GPU, and we used 8 GPUs for training. The initial base learning rate is 1×10^{-5} for backbone, 2.5×10^{-5} for text encoder, 1×10^{-5} for linear projection layers of Deformable DETR, and 1×10^{-4} for the rest of parameters. And we followed the linear learning rate scaling rule: $lr = base\_lr \times batch\_size/16$, where $batch\_size$ is the overall batch size of fully-annotated data. The text encoder uses linear learning rate decay with warmup schedule, and the rest uses step learning rate decay, with learning rate dropped after the 8th epoch. All parameters are optimized by Adam with weight decay of 1×10^{-4}. The pratraining of X-DETR with ResNet-101 backbone takes about 7 days on 8 GPUs for 10 epochs. The loss weights $\lambda_{cls} = 1$, $\lambda_{iou} = 2$, $\lambda_{L1} = 5$, and are set to 1 for object-phrase and image-caption alignment losses. The inference model is exponential moving averaged (EMA) from the model trajectory during training with a decay rate of 0.9998.



(a) Example of *mixed* (b) Example of *mixed** (c) Example of *mixed**+boxes

Fig. A. An image example for *mixed* (a), *mixed** (b) and *mixed**+boxes (c) of Table 6 in the paper. The text query is shown on the top left corner of each image. The text description on each bounding box is the noun phrase extracted from the text query. In the original *mixed* dataset of MDETR, the independent queries are merged into a single query (one sentence) and objects are sparsely annotated (annotations for the right two horses are missing), as in (a). We at first split the paragraph query into independent queries (two sentences) as in (b). Then we add COCO bounding boxes (right two horses) to the dataset withouht category information (no text description on the added bounding boxes), as in (c).

B.2 Finetuning on LVIS

When finetuning on LVIS, X-DETR was trained for 50 epochs for 1%/10%/100% data with batch size of 4 on each GPU. The initial base learning rate is 1×10^{-5} for backbone, 1×10^{-5} for text encoder, 5×10^{-6} for linear projection layers of Deformable DETR, and 5×10^{-5} for the rest of parameters. The learning rate dropped after the 40th epoch for step learning rate schedule. The image is resized such that the minimum of width and height is 800. The other settings are the same as pretraining. We used the category names as the language description of the object, but remove the text in the parentheses, e.g., "flip-flop_(sandal)" to "flip-flop".

B.3 Finetuning on Flickr30k

When finetuning on Flickr30k, X-DETR was trained for 3 epochs, with batch size of 4 on each GPU. The initial base learning rate is 5×10^{-6} for backbone, 5×10^{-6} for text encoder, 2.5×10^{-6} for linear projection layers of Deformable DETR, and 2.5×10^{-5} for the rest of parameters. The learning rate dropped after the 2nd epoch for step learning rate schedule. The other settings are the same as pretraining.

B.4 Finetuning on REC Datasets

When finetuning on REC datasets, we merged the RefCOCO/RefCOCO+/RefCOCOg together, excluding all images in all three validation sets. X-DETR was trained for 4 epochs, with batch size of 4 on each GPU. The initial base learning rate is 1×10^{-6} for for backbone, 1×10^{-5} for text encoder, 5×10^{-6} for linear projection layers of Deformable DETR, and 5×10^{-5} for the rest of parameters. The learning rate dropped after the 3rd epoch for step learning rate schedule. The other settings are the same as pretraining.

(a) (b) (c) (d)

Fig. B. The top images are pseudo label examples of LocNar, and the bottom images are the examples after adding OpenImage bounding boxes without category information (no text description on the added bounding boxes). The query lists are as follows,
(a): ['there is a food item on the plate.']
(b): ['This is an outside view.', 'At the bottom, the grass.', 'In the middle of the image there is a sea.', 'In the background there are many trees.', 'At the top of the image the sky and clouds.']
(c): ['flower plants.']
(d): ['a watermelon and watermelon slices.']

C Dataset Details

We used the *mixed* dataset of MDETR, which is a combination of Flicker30k entities [9], RefCOCO/RefCOCO+/RefCOCOg [7,13], Visual Genome (VG) [5], and GQA [2]. A typical example is shown in Fig. A (a). It can be found the query is a paragraph of queries and some objects are not annotated. As mentioned in the paper, we at first split the text query to a list of independent queries, as shown in Fig. A (b). Next, we append the COCO [6] bounding box annotation (without category information) to the image, as shown in Fig. A (c).

To obtain the pseudo labeled data on LocNar [10], given an image and its corresponding query, at first we use Spacy[1] to extract the noun phrases which are possible objects in the text query. Then we treat the pseudo-labeling as a phrase grounding task, retrieving the bounding box that is most aligned with the noun phrase. The model used for pseudo-labeling is trained on "+CC" of Table 6 in the paper. Some pseudo labeled examples are shown in Fig. B (top row). It can be found that the pseudo labels are reasonably good. But they could be not accurate, especially when there are multiple objects present in the image for a single noun phrase. For example, in Fig. B (a) (top row), we can only localize a single food item but miss the others because we do not know how many food items in this image. In addition, the OpenImages [4] object annotations (without category information) were also added to LocNar similar to COCO, as shown in Fig. B (bottom row).

[1] https://spacy.io/

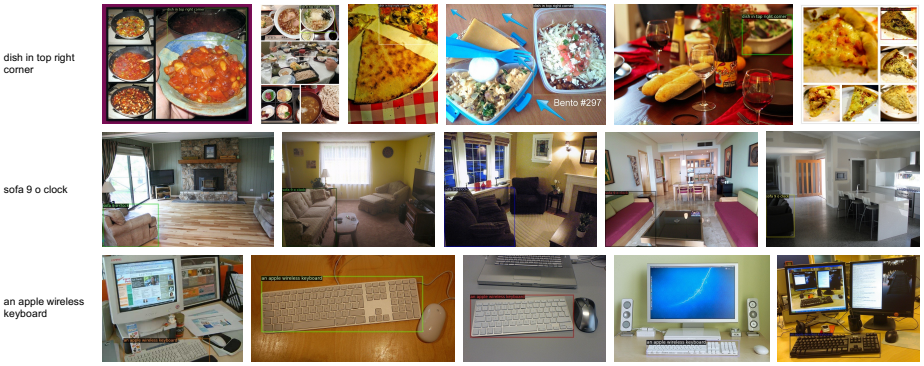


Fig. C. The top 5 MMIS retrieval results for the queries from RefCOCO/RefCOCO+/RefCOCOg (from top to bottom), on COCO validation set.

D MMIS Visualization

We have shown some examples of MMIS retrieval results on COCO dataset in Fig. C and on Objects365 [12] in Fig. D. It has shown that X-DETR can accurately retrieve the most relevant instance, with bounding box, to the query. The model can discriminate the differences between objects with different attributes. For example, the query of "skier in the sky" finds skiers jumping in the sky instead of standing on the snow. And "flying seagull" and "standing seagull" find seagull flying and standing, respectively. When given "red speedcar", the retrieved results are common red speedcars, e.g., in the parking lot or building. But when the attribute of "racing" is added, the most relevant results are speedcars in racing games. These have shown the power of X-DETR for any free-form language description. However, MMIS is still a very challenging task, and some of the top retrieved results may not be correct. For example, the third result for "skier in the sky" is mistaken due to the camera angle, and last result for "wet street" is wrong because of the building shadow. Also for "blue elephant", the top two results are wrong, probably because the model has never seen blue elephants during training, and the database may not have any true examples of "blue elephant". But interestingly, X-DETR does find two blue elephant statues, which could be the most relevant results in the database. MMIS is different from image-text retrieval, where the target of interest is at object-level instead of image-level. For example, the last image for "bottles on the shelf" is unlikely be retrieved by image-text retrieval, because those bottles only occupy a small portion of the whole image.

References

1. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV. pp. 213–229. Springer (2020)
2. Hudson, D.A., Manning, C.D.: Gqa: A new dataset for real-world visual reasoning and compositional question answering. In: CVPR. pp. 6700–6709 (2019)

3. Kamath, A., Singh, M., LeCun, Y., Synnaeve, G., Misra, I., Carion, N.: Mdetr-modulated detection for end-to-end multi-modal understanding. In: ICCV. pp. 1780–1790 (2021)
4. Krasin, I., Duerig, T., Alldrin, N., Ferrari, V., Abu-El-Haija, S., Kuznetsova, A., Rom, H., Uijlings, J., Popov, S., Veit, A., et al.: Openimages: A public dataset for large-scale multi-label and multi-class image classification. Dataset available from https://github. com/openimages **2**(3), 18 (2017)
5. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L., Shamma, D.A., Bernstein, M.S., Fei-Fei, L.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. Int. J. Comput. Vis. **123**(1), 32–73 (2017)
6. Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: Fleet, D.J., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV. vol. 8693, pp. 740–755. Springer (2014)
7. Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A.L., Murphy, K.: Generation and comprehension of unambiguous object descriptions. In: CVPR. pp. 11–20 (2016)
8. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
9. Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. Int. J. Comput. Vis. **123**(1), 74–93 (2017)
10. Pont-Tuset, J., Uijlings, J.R.R., Changpinyo, S., Soricut, R., Ferrari, V.: Connecting vision and language with localized narratives. In: ECCV. vol. 12350, pp. 647–664. Springer (2020)
11. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: CVPR. pp. 658–666 (2019)
12. Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., Sun, J.: Objects365: A large-scale, high-quality dataset for object detection. In: ICCV. pp. 8430–8439 (2019)
13. Yu, L., Poirson, P., Yang, S., Berg, A.C., Berg, T.L.: Modeling context in referring expressions. In: ECCV. vol. 9906, pp. 69–85. Springer (2016)

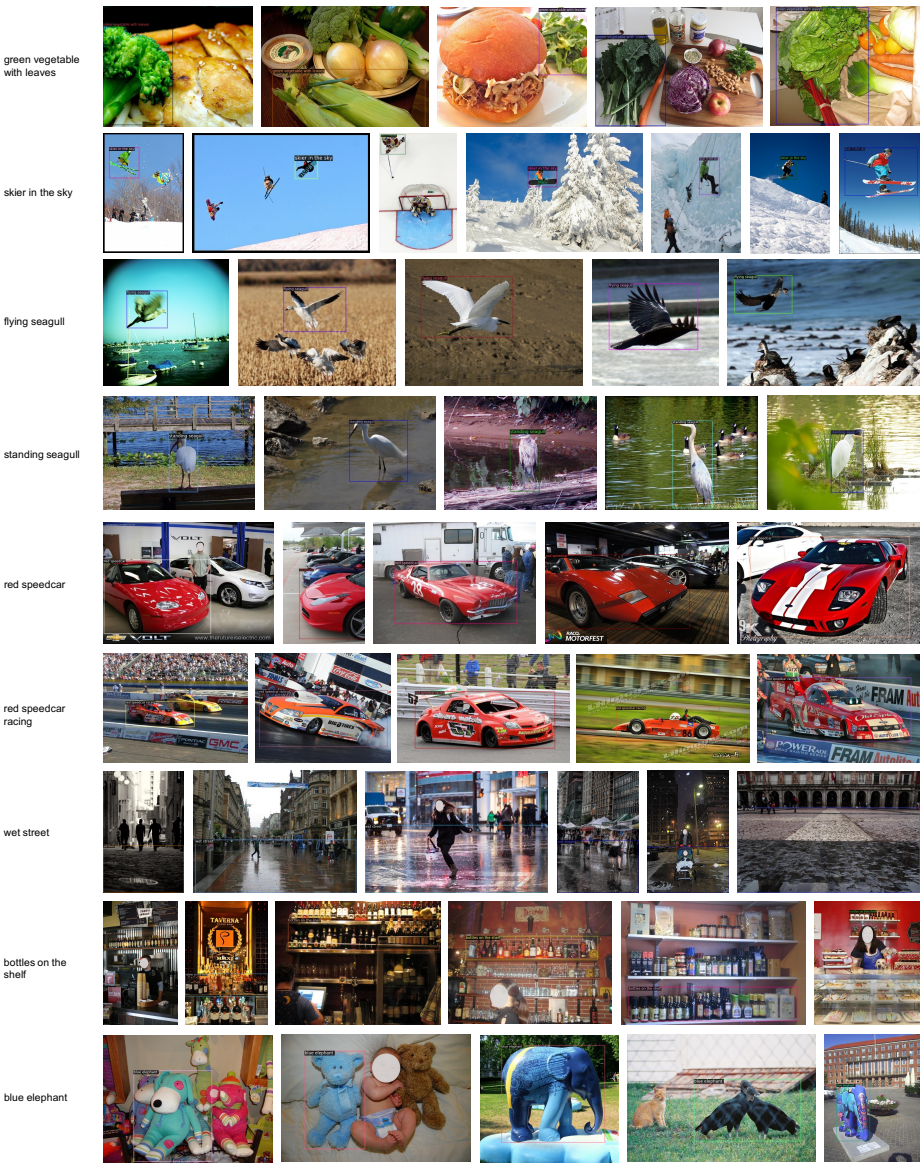


Fig. D. The top MMIS retrieval results for free-form language queries on the database of Objects365.