T-Rex2: Towards Generic Object Detection via Text-Visual Prompt Synergy

Qing Jiang[1,2], Feng Li[2,3], Zhaoyang Zeng[2], Tianhe Ren[2], Shilong Liu[2,4], and Lei Zhang[1,2]

[1] South China University of Technology
[2] International Digital Economy Academy (IDEA)
[3] The Hong Kong University of Science and Technology
[4] Tsinghua University
mountchicken@outlook.com , fliay@connect.ust.hk ,
lius120@mails.tsinghua.edu.cn
{rentianhe, zengzhaoyang, leizhang}@idea.edu.cn
https://deepdataspace.com/home

1 Model Details

1.1 Implementation Details

For the vision backbone, we use Swin Transformer [6] that is pre-trained on ImageNet [1]. For the text encoder, we use the text encoder from the open-sourced CLIP[5]. During Hungarian matching, we only use classification loss, box L1 loss, and GIOU [9] loss. The loss weights are 2.0, 5.0, and 2.0, respectively. For the final loss, we use classification loss, box L1 loss, GIOU loss, and the proposed constrastive alignment loss, and set the weights to be 1.0, 5.0, 2.0, and 1.0, respectively. Following DINO [12], we use contrastive denoising training (CDN) to stabilize training and accelerate convergence.

We use automatic mixed precision for training. For the Swin Transformer tiny model, the training is performed on 16 NVIDIA A100 GPUs with a total batch size of 128. For the Swin Transformer large model, the training is performed on 32 NVIDIA A100 GPUs with a total batch size of 64.

1.2 Inference Speed

In this section, we measure the inference speed of each module of *T-Rex2*. The experiment is conducted on an NVIDIA RTX 3090 GPU with a batch size of 1. Before measurement, we conducted a warm-up phase to stabilize GPU performance. Inference times were recorded over 100 iterations, ensuring accuracy through the use of `torch.cuda.synchronize()` to account for CUDA's asynchronous execution. The results are shown in Tab. 1. Benefiting from the late fusion design, *T-Rex2* can work in real-time when using the interactive visual prompt mode. Specifically, after a user uploads a picture, we only need to process

[5] https://github.com/openai/CLIP

Backbone	backbone	encoder	visual prompt encoder	text prompt encoder	box decoder	FPS	Interactive FPS
Swin-T	0.0318	0.0240	0.0120	0.0103	0.0180	10.41	33.33
Swin-L	0.1220	0.0929	0.0261	0.0116	0.0240	3.62	19.96

Table 1: Time cost for each module in *T-Rex2*. Interactive FPS is the inference speed of the visual prompt encoder and the box decoder. Since *T-Rex2* is a late fusion model, we only need to forward the backbone and encoder for once, and multi-round interactions only require running the prompt encoder and decoder

it once with our main processing steps (backbone and encoder) to get the image features. Any further interactions from the user involve just running our visual prompt encoder and decoder multiple times, which is in a real-time manner. This quick response is especially useful for scenarios like automatic annotation.

2 Data Engine Details

2.1 Text Prompt Data Engine

To collect region-text pairs from caption datasets LAION400M [10] and Conceptual Captions [11], We first use CLIP to compute the CLIP score for each image and its caption and retain only pairs of image descriptions with similarity greater than 0.8. Next, we use spaCy to extract the noun phrases in each caption and then use these nouns to prompt the GroundingDINO [5] model to get the box regions corresponding to these noun phrases in the image. Finally, we will compute the CLIP score for each box region and its corresponding noun phrases once more, and keep only the pairs with similarity greater than 0.8.

2.2 Image Prompt Data Engine

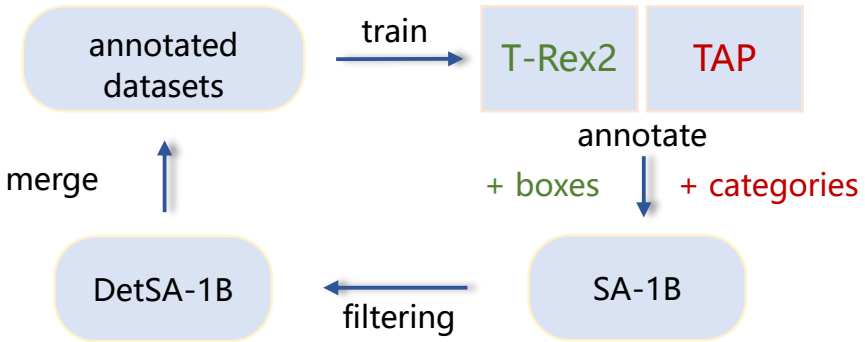


Fig. 1: Workflow of the proposed image prompt data engine.

Data Type	Dataset	# Images
Text prompt	Conceptual Captions	1,840,473
	LAION400M	1,202,245
	Bamboo	1,109,856
Visual Prompt	SA-1B	653,285

Table 2: Data statistics of data collected from text prompt and visual prompt engines

We show the overview of the proposed data engine in Fig. 1 and some examples in DetSA-1B in Fig. 2.

2.3 Data Statistics

We list the amount of data collected from the two data engines in Tab. 2.

Method	Backbone	COCO-Val Zero-Shot		LVIS-Val Zero-Shot	
		Acc@Top1	Acc@Top5	Acc@Top1	Acc@Top5
CLIP	ViT-B	37.6	60.1	9.0	20.0
T-Rex2	Swin-T	72.6	89.4	40.8	67.5
T-Rex2	Swin-L	**82.2**	**93.9**	**49.8**	**76.9**

Table 3: Zero-shot region classification results. For each dataset, we use its full categories as the classification target and calculate the Top1 and Top5 classification accuracy. For CLIP, we crop the region out for classification.

3 Advanced Capabilities of *T-Rex2*

3.1 Region Classification

The contrastive alignment between text prompts and visual prompts also unlocks the capability to classify regions for visual prompts. Much like the zero-shot classification approach of CLIP, we can assign category labels to visual prompts by measuring the similarity between visual prompts and pre-computed text prompts:

$$\text{Label} = \text{argmax}\, j \left(\frac{\exp(V \cdot t_j)}{\sum_{l=1}^{K} \exp(V \cdot t_l)} \right) \tag{1}$$

We can use predefined category names to pre-compute the text embeddings which enable us to identify arbitrary objects through visual prompting.

We show the zero-shot region classification results on COCO [4] and LVIS [2] in Tab. 3. We use each GT box as the visual prompt and compute the similarity with all the category names in that dataset. Compared to CLIP, *T-Rex2* possesses stronger region classification capability. We show some visualization results in Fig. 3.

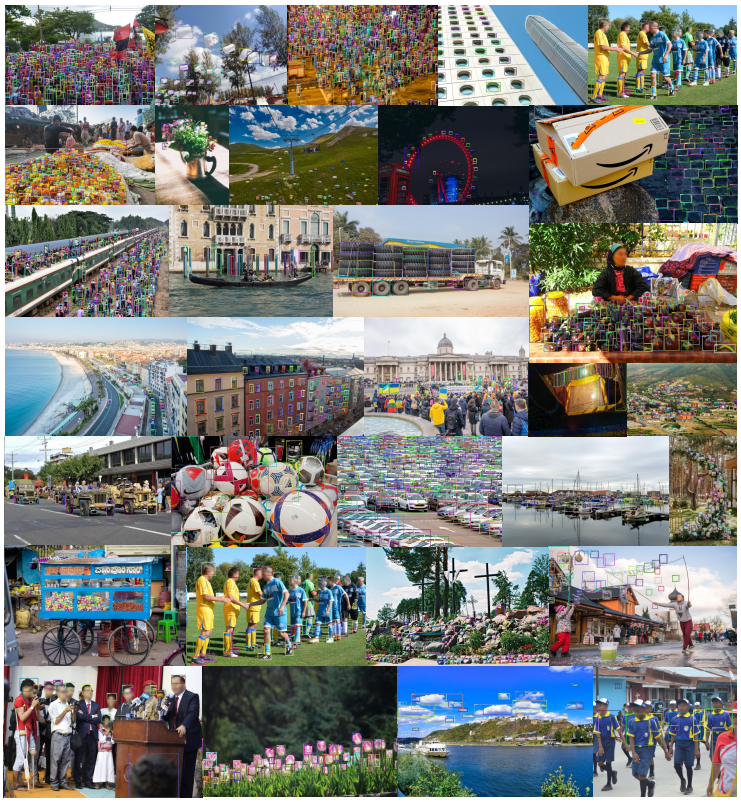


Fig. 2: Image examples in DetSA-1B.

3.2 Open-set Video Object Detection

T-Rex2 can also be used for open-set video object detection. Given a video, we can extract any N frames, customize a generic visual embedding for a certain object by using *T-Rex2*'s generic visual prompt workflow, and then use this embedding to detect all frames in the video. We also show some visualization results in Fig. 4. Despite not being trained on video data, *T-Rex2* can also detect objects in videos well.

4 More Experiment Details

4.1 Details on Object Counting Task

We evaluate *T-Rex2* on the object counting task to show its interactive object detection capability. Specifically, we are focusing on the few-shot object counting task. In this task, each image will be provided with three exemplar boxes on the current image to indicate the target object and require the output of the number of the target object.

Fig. 3: Visualization results of region classification workflow. We use a dictionary of 2560 classes to classify the visual prompts. The classification result is shown at the bottom right for each image.

Settings. We conduct evaluations on the commonly used counting dataset FSC147 [8]and the more challenging dataset FSCD-LVIS [7]. FSC147 comprises 147 categories of objects and 1190 images in the test set and FSCD-LVIS comprises 377 categories and 1014 images in the test set. Both two datasets provide three bounding boxes of exemplar objects for each image, which we will use as the visual prompt for *T-Rex2*.

Metric. We adopt the Mean Average Error (MAE) metric, a widely employed standard in object counting. The mathematical expression is as follows:

$$\text{MAE} = \frac{1}{J}\sum_{j=1}^{J}\left|c_j^* - c_j\right| \tag{2}$$

We report MAE on the FSC147 dataset as it doesn't provide ground truth boxes on test set images, and report AP on the FSCD-LVIS dataset as it provides ground truth boxes. We show some prediction results of *T-Rex2* on the FSC147 and FSCD datasets in Fig. 5.

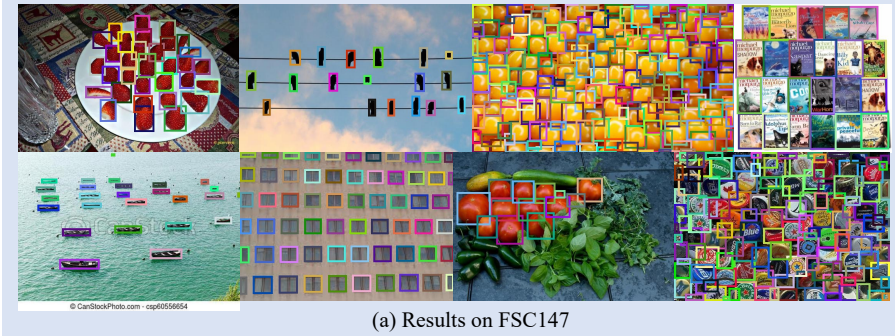
4.2 Visualization Comparison with T-Rex

In Fig. 6, we compare the detection results between T-Rex [3] and *T-Rex2*. In interactive visual prompt detection mode, both models exhibit comparable performance in single-object scenes (where there is no interference from other objects in the image). For multi-object scenarios, T-Rex is more prone to false detections, whereas *T-Rex2* exhibits fewer false detections, indicating a better distinction between objects. This improvement is attributed to the joint training with text and visual prompts. For generic visual prompt detection mode, *T-Rex2* also shows more advantages.

Fig. 4: *T-Rex2* on zero-shot video object detection task. We randomly sample 4 frames from a given video and customize a generic visual embedding for an object through the generic visual prompt workflow of *T-Rex2*. This visual embedding will be used for inference for all video frames.

5 Limitations

Despite the integration of text and visual prompts showing mutual benefits within a unified model, challenges arise. Visual prompts may sometimes interfere with text prompts, especially in scenarios involving common objects, as indicated by the reduced performance on the COCO benchmark when both are used together. Despite this, improvements on the LVIS benchmark highlight the potential benefits of this approach. Therefore, further research into improving the alignment between these modalities is essential. Moreover, the requirement for up to 16 visual examples to ensure reliable detection due to visual diversity highlights a need for methods that enable visual prompts to achieve similar effectiveness with fewer visual examples.



(a) Results on FSC147

(b) Results on FSCD-LVIS

Fig. 5: The prediction results of *T-Rex2* on the FSC147 and FSCD-LVIS datasets, respectively.

References

1. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
2. Gupta, A., Dollar, P., Girshick, R.: Lvis: A dataset for large vocabulary instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5356–5364 (2019)
3. Jiang, Q., Li, F., Ren, T., Liu, S., Zeng, Z., Yu, K., Zhang, L.: T-rex: Counting by visual prompting. arXiv preprint arXiv:2311.13596 (2023)
4. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)
5. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023)
6. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022 (2021)

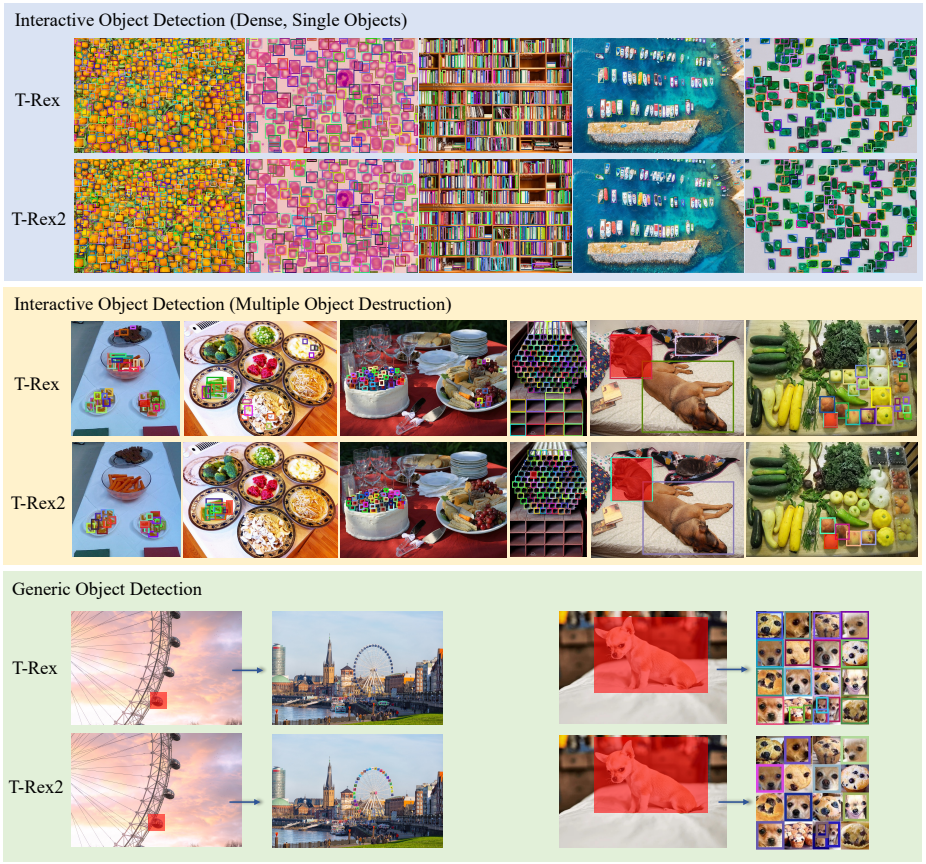


Fig. 6: Visualization comparison between T-Rex and *T-Rex2*.

7. Nguyen, T., Pham, C., Nguyen, K., Hoai, M.: Few-shot object counting and detection. In: European Conference on Computer Vision. pp. 348–365. Springer (2022)
8. Ranjan, V., Sharma, U., Nguyen, T., Hoai, M.: Learning to count everything. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3394–3403 (2021)
9. 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: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 658–666 (2019)
10. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: An open large-scale dataset for training next generation image-text models. arXiv preprint arXiv:2210.08402 (2022)
11. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2556–2565 (2018)

12. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L.M., Shum, H.Y.: Dino: Detr with improved denoising anchor boxes for end-to-end object detection (2022)