Unified Visual Relationship Detection with Vision and Language Models: Supplementary Material

Long Zhao Liangzhe Yuan Boqing Gong Yin Cui Florian Schroff
Ming-Hsuan Yang Hartwig Adam Ting Liu

Google Research

{longzh,liuti}@google.com

1. Implementation Details

1.1. Architecture

Vision and language model (VLM): We adopt the pre-trained CLIP [23] and LiT [32], which both provide open-source implementations and pre-trained checkpoints. We experiment with CLIP backbones of ViT-B/32, ViT-B/16, and ViT-L/14, and LiT backbones of ViT-B/32, R26+B/1, and ViT-H/14. These backbones follow the nomenclature from [3] for model size, patch size, and Transformer *vs*. hybrid architectures. For example, B/32 refers to ViT-Base with patch size 32, while R26+B/1 refers to a hybrid ResNet-26 plus ViT-Base with stride 1. We use the pre-trained CLIP weights provided by the original paper and obtain the pre-trained LiT weights from OWL-ViT [21].

Object detector: To adapt VLMs for object detection, we remove token pooling and add detection heads, which contain one linear layer for producing classification embeddings and the other two-layer feed-forward network for box prediction. Following the design of OWL-ViT [21], we add a bias to the predicted box coordinates so that each box is by default centered on the image patch that corresponds to the token from which this box is predicted when arranging the token sequence as a 2D grid. Although the stochastic depth regularisation [10] (*i.e*., droplayer) is not applied during VLM pre-training, we add it to image encoders during fine-tuning, which reduces model overfitting. In addition, we merge the class token into other feature map tokens by multiplying it with them, and append layer norm to the output of CLIP models, following the practice in [21].

Relationship decoder: We use Perceiver Resampler proposed in Flamingo [1] as our decoder backbone. It contains three Transformer layers with eight attention heads. Note that we employ ReLU instead of Squared ReLU used in Flamingo to simplify the design. We set the number of input relation queries to 100 for all datasets. Each Transformer layer is implemented following the post-normalization design [27], which requires a linear warm-up learning rate schedule ($1,000$ warm-up steps) for model training.

1.2. Training Object Detector

We list the hyper-parameters used for training our object detector in Table 1. Its training procedure follows [21], except that we make two major modifications. First, we freeze the text encoder of a pre-trained VLM. This is because we would like to keep the embedding space of a pre-trained text encoder so that its discriminative capability is still sufficient for encoding relationship triplets, preventing it from forgetting issues. Second, we apply the stochastic depth regularisation [10] (*i.e*., droplayer) to image encoders when using LiT [32] as the pre-trained VLM; otherwise, performance drops will be observed. When training CLIP models, we mix the COCO [18], Objects365 [24], HICO-DET [2], and Visual Genome [14] datasets randomly in each batch with probabilities of 0.1, 0.5, 0.2, and 0.2, respectively; when training LiT models, we use probabilities of 0.1, 0.7, 0.1, and 0.1, respectively. This is due to the fact that LiT models suffer from overfitting if a larger probability is applied to HICO-DET or Visual Genome.

1.3. Training Relationship Decoder

In the experiments, we introduce two configurations for training our relationship decoders: dataset-specific models and unified models. They use different training setups which are shown in Table 2.

Dataset-specific models: To reduce model overfitting, we fix both the text encoder and object detector when training dataset-specific models. We train all models in this configuration with $140,000$ steps, the learning rate of 1×10^{-4}, and batch size 64. When training dataset-specific models on the V-COCO dataset, we observe serious overfitting problems as its training set is extremely small (less than $5,000$ images). Therefore, we early stop the model training, where we use at most $20,000$ training steps, while keeping other hyper-parameters unchanged.

Model	Backbone	# of steps	Batch size	Learning rate	Droplayer rate	Image size	Dataset proportions	Frozen text
UniVRD (CLIP)	ViT-B/32	$140,000$	256	5×10^{-5}	0.2	768	$0.1/0.5/0.2/0.2$	✓
UniVRD (CLIP)	ViT-B/16	$140,000$	256	5×10^{-5}	0.2	768	$0.1/0.5/0.2/0.2$	✓
UniVRD (CLIP)	ViT-L/14	$70,000$	256	2×10^{-5}	0.2	672	$0.1/0.5/0.2/0.2$	✓
UniVRD (LiT)	ViT-B/32	$140,000$	256	2×10^{-4}	0.2	768	$0.1/0.7/0.1/0.1$	✓
UniVRD (LiT)	R26+B/1	$140,000$	256	2×10^{-4}	0.2	768	$0.1/0.7/0.1/0.1$	✓
UniVRD (LiT)	ViT-H/14	$70,000$	256	5×10^{-5}	0.2	480	$0.1/0.7/0.1/0.1$	✓

Table 1. **List of hyper-parameters used for training our object detector.** The mix probabilities of the COCO [18], Objects365 [24], HICO-DET [2], and Visual Genome [14] datasets within each batch are shown in dataset proportions. Note that we only apply stochastic depth regularisation [10] (*i.e.*, droplayer) to image encoders, as text encoders are frozen.

Model	Backbone	# of steps	Batch size	Learning rate	Droplayer rate	Image size	Dataset proportions	Frozen text
Dataset-specific models								
UniVRD (CLIP)	ViT-B/32	$140,000$	64	1×10^{-4}	0.0	768	-	✓
UniVRD (CLIP)	ViT-B/16	$140,000$	64	1×10^{-4}	0.0	768	-	✓
UniVRD (CLIP)	ViT-L/14	$140,000$	64	1×10^{-4}	0.0	672	-	✓
Unified models								
UniVRD (CLIP)	ViT-B/32	$140,000$	256	$1 \times 10^{-4}/2 \times 10^{-6}$	0.2	768	$0.5/0.1/0.4$	✗
UniVRD (CLIP)	ViT-B/16	$140,000$	256	$1 \times 10^{-4}/2 \times 10^{-6}$	0.2	768	$0.5/0.1/0.4$	✗
UniVRD (CLIP)	ViT-L/14	$140,000$	256	$1 \times 10^{-4}/2 \times 10^{-6}$	0.2	672	$0.5/0.1/0.4$	✗

Table 2. **List of hyper-parameters used for training our visual relationship decoder.** Where two numbers are given for the learning rate, the first is for the visual relationship decoder and the second for the rest of the whole model. The mix probabilities of the HICO-DET [2], V-COCO [8], and Visual Genome [14] datasets within each batch are shown in dataset proportions. Note that we only apply stochastic depth regularisation [10] (*i.e.*, droplayer) to image encoders.

Unified models: When training unified models, we mix HICO-DET, V-COCO, and Visual Genome randomly in each batch with probabilities of 0.5, 0.1, and 0.4, respectively. We find that further increasing the mix ratio of training data from Visual Genome will lead to model overfitting on this dataset. We enlarge the batch size to 256 and set the learning rate of both the text encoder and object detector to 2×10^{-6}. We have also tried the same optimization setup for training dataset-specific models, but it will lead to performance drops. This suggests the benefits of enlarging the batch size and unfreezing pre-trained models when we train unified models across multiple datasets.

2. Additional Results

2.1. Mosaics Image Augmentation

Table 3 shows our results on HICO-DET [2] when different mosaics configurations are employed for training the visual relationship decoder. To highlight the performance differences, we report the results without using per-class PNMS. We can find that using only 1×1 single images is clearly worse than including larger mosaics (*i.e.*, smaller mosaic tiles), and the model achieves the best performance with the inclusion of 3×3 mosaics.

2.2. Ablation on Training Unified Models

We identify the top three important factors affecting the performance of our unified models in Table 4. We can find that using a cascade training paradigm still leads to a substantial performance boost, which is consistent with our observations on training dataset-specific models. In contrast, freezing either the object detector or the text encoder when training the relationship decoder causes performance drops.

Mosaics ratio			Default (%)		
1×1	2×2	3×3	mAP_F	mAP_R	mAP_N
1.0	0.0	0.0	25.84	20.09	27.56
0.6	0.4	0.0	27.06	19.66	29.27
0.4	0.3	0.3	27.92	20.98	30.00

Table 3. **Performance comparison when different mosaics ratios are utilized for image augmentation.** We report the results of UniVRD (CLIP) using the ViT-B/32 backbone on the HICO-DET test set without performing per-class PNMS.

Ablation	mAP_F
Unified baseline	29.47
(1) Use one-stage training schedule	-4.97
(2) Freeze the object detector in the second stage	-1.21
(3) Freeze the text encoder in the second stage	-0.83

Table 4. **Ablation study of the main methodological improvements for training unified models.** For simplicity, difference in mAP to the *unified baseline* is shown. All ablations are carried out for the UniVRD (CLIP: ViT-B/32) model on HICO-DET.

Method	HOTR	QPIC	**UniVRD**
HICO-DET (mAP_F)	25.1	29.1	**29.7**
Visual Genome (mR@50)	9.4	-	**9.6**

Table 5. **Results with the ResNet-50 backbone on HICO-DET and VG.** Our model is initialized from CLIP [23].

This is because of the fact that our proposed unified VRD framework makes it possible for us to train models with a larger amount of data across multiple datasets at the same time, mitigating the model overfitting issue. Hence, we are able to train models with larger learning capabilities (with more model parameters to be fine-tuned).

2.3. Results with Different Backbones

To conduct a fair comparison on the backbone, we show results of different models using the same ResNet-50 backbone in Table 5. We can observe that our method achieves competitive performances on both HICO-DET and VG.

2.4. HOI Detection on V-COCO

In this section, we provide additional results on the V-COCO dataset [8]. The metric of role AP is used for evaluation: a detection is correct if the location of the agent (*i.e.*, both subjects and objects) and each role (*i.e.*, predicate classes) is correct (correctness is measured using bounding box overlap as is standard). In V-COCO, there are a number of HOI categories which are defined with no object labels. To deal with this situation, we evaluate the model performance in two different scenarios following the official evaluation scheme of V-COCO. In Scenario 1 ($AP^{S\#1}_{role}$), detectors are required to report cases in which there is no object, while in Scenario 2 ($AP^{S\#2}_{role}$), we just ignore the prediction of an object bounding box in these cases.

To deal with the long-tail class distribution in V-COCO, we use the dynamic re-weighting [33] during model training. To handle HOI categories which do not contain objects, we conduct the following modifications to let them be compatible with the proposed framework. First, for each sample, their subject annotations are employed as pseudo ground-truth objects to be predicted by our model. Second, we use the prompt template 'a ⟨subject⟩ ⟨predicate⟩-ing' for HOI categories including transitive verbs and the prompt template 'a ⟨subject⟩ ⟨predicate⟩-ing something' for those containing intransitive verbs.

We compared the proposed method with both bottom-up and single-stage methods. As illustrated in Table 6, we can find that (1) we are able to achieve the state-of-the-art performance; (2) our model outperforms other bottom-up approaches by a significant margin; (3) further improvements can be obtained when we scale up the model. These observations are consistent with our results on HICO-DET, which re-confirm the effectiveness of the proposed method.

Model	Extra-sup.	$AP^{S\#1}_{role}$	$AP^{S\#2}_{role}$
Single-stage methods			
UnionDet [11]	✗	47.5	56.2
HOI-Transformer [36]	✗	52.9	-
GGNet [35]	✗	54.7	-
HOTR [12]	✗	55.2	64.4
DIRV [4]	✗	56.1	-
QPIC [25]	✗	58.8	61.0
CDN [33]	✗	61.7	63.8
RLIP [31]	VG†	61.9	64.2
GEN-VLKT [17]	CLIP†	**62.4**	**64.5**
Bottom-up methods			
InteractNet [7]	✗	40.0	-
GPNN [22]	✗	44.0	-
iCAN [6]	✗	45.3	52.4
TIN [16]	✗	47.8	54.2
VCL [9]	✗	48.3	-
DRG [5]	Text [20]	51.0	-
IP-Net [30]	✗	51.0	-
VSGNet [26]	✗	51.8	57.0
PMFNet [28]	Pose [18]	52.0	-
PD-Net [34]	Text [20]	52.6	-
CHGNet [29]	✗	52.7	-
FCMNet [19]	Text [20]	53.1	-
ACP [13]	Text [20]	53.2	-
IDN [15]	✗	53.3	60.3
UniVRD (CLIP: ViT-B/32)	CLIP†	59.9	62.7
UniVRD (CLIP: ViT-B/16)	CLIP†	62.3	64.8
UniVRD (CLIP: ViT-L/14)	CLIP†	65.1	66.3
UniVRD (LiT: ViT-B/32)	LiT†	59.4	62.2
UniVRD (LiT: R26+B/1)	LiT†	62.6	65.1
UniVRD (LiT: ViT-H/14)	LiT†	**65.8**	**66.9**

Table 6. **System-level comparison on V-COCO.** † denotes training supervisions obtained from the model pre-training stage. Best performances are highlighted in bold.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: A visual language model for few-shot learning. In *NeurIPS*, 2022.

[2] Yu-Wei Chao, Yunfan Liu, Xieyang Liu, Huayi Zeng, and Jia Deng. Learning to detect human-object interactions. In *WACV*, 2018.

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[4] Hao-Shu Fang, Yichen Xie, Dian Shao, and Cewu Lu. DIRV:

Dense interaction region voting for end-to-end human-object interaction detection. In *AAAI*, 2021.

[5] Chen Gao, Jiarui Xu, Yuliang Zou, and Jia-Bin Huang. DRG: Dual relation graph for human-object interaction detection. In *ECCV*, 2020.

[6] Chen Gao, Yuliang Zou, and Jia-Bin Huang. iCAN: Instance-centric attention network for human-object interaction detection. In *BMVC*, 2018.

[7] Georgia Gkioxari, Ross Girshick, Piotr Dollár, and Kaiming He. Detecting and recognizing human-object interactions. In *CVPR*, 2018.

[8] Saurabh Gupta and Jitendra Malik. Visual semantic role labeling. *arXiv preprint arXiv:1505.04474*, 2015.

[9] Zhi Hou, Xiaojiang Peng, Yu Qiao, and Dacheng Tao. Visual compositional learning for human-object interaction detection. In *ECCV*, 2020.

[10] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016.

[11] Bumsoo Kim, Taeho Choi, Jaewoo Kang, and Hyunwoo J Kim. UnionDet: Union-level detector towards real-time human-object interaction detection. In *ECCV*, 2020.

[12] Bumsoo Kim, Junhyun Lee, Jaewoo Kang, Eun-Sol Kim, and Hyunwoo J Kim. HOTR: End-to-end human-object interaction detection with transformers. In *CVPR*, 2021.

[13] Dong-Jin Kim, Xiao Sun, Jinsoo Choi, Stephen Lin, and In So Kweon. Detecting human-object interactions with action co-occurrence priors. In *ECCV*, 2020.

[14] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael S Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017.

[15] Yong-Lu Li, Xinpeng Liu, Xiaoqian Wu, Yizhuo Li, and Cewu Lu. HOI analysis: Integrating and decomposing human-object interaction. In *NeurIPS*, 2020.

[16] Yong-Lu Li, Siyuan Zhou, Xijie Huang, Liang Xu, Ze Ma, Hao-Shu Fang, Yanfeng Wang, and Cewu Lu. Transferable interactiveness knowledge for human-object interaction detection. In *CVPR*, 2019.

[17] Yue Liao, Aixi Zhang, Miao Lu, Yongliang Wang, Xiaobo Li, and Si Liu. GEN-VLKT: Simplify association and enhance interaction understanding for HOI detection. In *CVPR*, 2022.

[18] 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 *ECCV*, 2014.

[19] Yang Liu, Qingchao Chen, and Andrew Zisserman. Amplifying key cues for human-object-interaction detection. In *ECCV*, 2020.

[20] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[21] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. Simple open-vocabulary object detection with vision transformers. In *ECCV*, 2022.

[22] Siyuan Qi, Wenguan Wang, Baoxiong Jia, Jianbing Shen, and Song-Chun Zhu. Learning human-object interactions by graph parsing neural networks. In *ECCV*, 2018.

[23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[24] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *ICCV*, 2019.

[25] Masato Tamura, Hiroki Ohashi, and Tomoaki Yoshinaga. QPIC: Query-based pairwise human-object interaction detection with image-wide contextual information. In *CVPR*, 2021.

[26] Oytun Ulutan, ASM Iftekhar, and Bangalore S Manjunath. VSGNet: Spatial attention network for detecting human object interactions using graph convolutions. In *CVPR*, 2020.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[28] Bo Wan, Desen Zhou, Yongfei Liu, Rongjie Li, and Xuming He. Pose-aware multi-level feature network for human object interaction detection. In *ICCV*, 2019.

[29] Hai Wang, Wei-shi Zheng, and Ling Yingbiao. Contextual heterogeneous graph network for human-object interaction detection. In *ECCV*, 2020.

[30] Tiancai Wang, Tong Yang, Martin Danelljan, Fahad Shahbaz Khan, Xiangyu Zhang, and Jian Sun. Learning human-object interaction detection using interaction points. In *CVPR*, 2020.

[31] Hangjie Yuan, Jianwen Jiang, Samuel Albanie, Tao Feng, Ziyuan Huang, Dong Ni, and Mingqian Tang. RLIP: Relational language-image pre-training for human-object interaction detection. In *NeurIPS*, 2022.

[32] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. LiT: Zero-shot transfer with locked-image text tuning. In *CVPR*, 2022.

[33] Aixi Zhang, Yue Liao, Si Liu, Miao Lu, Yongliang Wang, Chen Gao, and Xiaobo Li. Mining the benefits of two-stage and one-stage HOI detection. In *NeurIPS*, 2021.

[34] Xubin Zhong, Changxing Ding, Xian Qu, and Dacheng Tao. Polysemy deciphering network for human-object interaction detection. In *ECCV*, 2020.

[35] Xubin Zhong, Xian Qu, Changxing Ding, and Dacheng Tao. Glance and gaze: Inferring action-aware points for one-stage human-object interaction detection. In *CVPR*, 2021.

[36] Cheng Zou, Bohan Wang, Yue Hu, Junqi Liu, Qian Wu, Yu Zhao, Boxun Li, Chenguang Zhang, Chi Zhang, Yichen Wei, et al. End-to-end human object interaction detection with HOI transformer. In *CVPR*, 2021.