Supplementary Materials for Referring Atomic Video Action Recognition

Kunyu Peng[1,*], Jia Fu[3,4,*], Kailun Yang[2,†], Di Wen[1], Yufan Chen[1], Ruiping Liu[1], Junwei Zheng[1], Jiaming Zhang[1], M. Saquib Sarfraz[1,6], Rainer Stiefelhagen[1], and Alina Roitberg[5]

[1]Karlsruhe Institute of Technology, [2]Hunan University,
[3]RISE Research Institutes of Sweden, [4]KTH Royal Institute of Technology,
[5]University of Stuttgart, [6]Mercedes-Benz Tech Innovation

A Potential Impacts

In this section, we deliberate on the societal implications of our research endeavors. We introduce a new task, Referring Atomic Video Action Recognition (RAVAR), by developing the RefAVA dataset and establishing the RAVAR benchmark. A total of 36,630 instances were meticulously annotated based on videos sourced from the AVA dataset to facilitate subsequent inquiries into the RAVAR domain. In contrast to traditional approaches in atomic video action detection, which typically depend on post-hoc selections of the person of interest, our methodology leverages textual descriptions as indicators. This approach mitigates the reliance on precise positional data delineated by bounding boxes, which may exhibit significant variability over time.

The introduction of this task setting is crucial for advancing deep learning models' capabilities in comprehending scenes with greater details, particularly within the atomic video action recognition framework. Such enhanced understanding is challenging and important in scenarios involving complex interactions among multiple individuals, as is often encountered in fields such as rehabilitation and robotic assistance. This shift towards a more nuanced understanding has the potential to significantly impact these applications, offering more adaptable and context-aware solutions.

To construct the first testbed for the RAVAR task, we use 11 well-established methods from Atomic Action Localization (AAL), Text Video Retrieval (TVR), and Video Question Answering (VQA) domains while observing that most of the selected baselines cannot deliver satisfactory performances. AAL approaches are not good at dealing with the referring location of the person of interest while TVR and VQA approaches are not good at handling the fine-grained action recognition. We propose `RefAtomNet` to address both of the aforementioned issues. `RefAtomNet` relies on three streams of token extraction, namely the textual reference tokens, the visual tokens, and the newly proposed location-semantic tokens, which are used for incorporating semantic and location cues provided

* Equal contribution
† Correspondence: kailun.yang@hnu.edu.cn

by well-established object detectors from the scene. `RefAtomNet` further takes advantage of the 1D sequential agent attention on each stream of the tokens to achieve self-suppression regarding the irrelevant tokens. It then utilizes the agent-based location semantic aware attentional fusion to enhance the visual stream by merging the agent attention masks and the agent tokens from the textual reference and the location-semantic streams. The proposed `RefAtomNet` achieves state-of-the-art on the RAVAR benchmark, with great generalizability towards the visual textual backbone and object detector. `RefAtomNet` can also deliver promising RAVAR performances when faced with the test time reference rephrasing and test time video disturbances compared with the chosen baselines, which benefits practical deployment. However, our method still has the potential to output false predictions and biased content, which can have undesired consequences, impacting society negatively.

B Discussion of Limitations

RAVAR by referring to multiple persons preserving the same attributes though one sentence in one video is not tackled in our work, since we annotate each individual and ensure that the described person can be successfully referred by the provided description, and there are no two persons who share the same reference sentence in one video clip. However, we regard this direction as an interesting future work direction for the RAVAR task. Since we deliver the first dataset for the RAVAR field and there is no other existing dataset for this new task, our experiments are only conducted on the contributed RefAVA dataset.

C Evaluation Metrics

C.1 Mean Average Precision (mAP)

The mAP for multi-label classification measures prediction precision across labels by computing precision at different thresholds and plotting precision-recall curves for each label. The Average Precision (AP) for each label is derived by integrating over these curves, typically approximated by summing areas under specific points. The mAP, an average of these AP values across all labels, serves as an aggregate performance metric for accurately predicting multiple labels per instance. This aggregate performance measure reflects the model's proficiency in accurately predicting multiple labels for each instance within the dataset. It accounts for the model's ability not only to identify the presence of various labels but also to ascertain their absence, thereby ensuring a balanced evaluation of its predictive capabilities across all possible label outcomes.

C.2 Area Under the Receiver Operating Characteristic (AUROC)

The AUROC for multi-label classification is determined by considering each label as a separate binary classification and calculating the True Positive Rate

(TPR) and False Positive Rate (FPR) to plot ROC curves for each label. The overall AUROC for the multi-label classification model is then calculated by averaging the individual AUROC scores across all labels. This aggregated metric, often referred to as the macro-average AUROC, provides a global indicator of the model's discriminative capability across the entire multi-label task. It is a critical measure, particularly where the balance between different classes varies significantly, as it offers an unbiased metric that does not favor labels with more instances. Emphasizing the importance of AUROC in multi-label classification highlights its role in ensuring the model's robustness and effectiveness across diverse conditions.

C.3 Mean Intersection of Union (mIOU)

The mIOU is a prominent evaluation measure used in the context of bounding box regression tasks, particularly in object detection. We use mIOU to evaluate the bounding box regression ability for the person of interest. This metric is chosen as an auxiliary indicator that has less priority than the mAP and the AUROC metrics since our main aim is to harvest the correct atomic action predictions for the referring person.

C.4 Further Clarification regarding the Metrics

Our task prioritizes recognition over localization to progressively address this new RAVAR challenge. We evaluate detection and recognition performance separately, treating mIOU as a secondary metric. We provide mAP (IOU=0.2, 0.5) of our approach and BLIPv2, which delivers the best performances among all leveraged baselines, as follows.

	BLIPv2-Val	**BLIPv2-Test**	**Ours-Val**	**Ours-Test**
mAP (IOU=0.2)	46.77	45.21	**51.26**	**49.47**
mAP (IOU=0.5)	41.40	39.27	**44.84**	**41.93**

D Generalizability to Different Detectors

In this work, we also deliver the ablation towards how much will the object detector affect the performance of RAVAR in Tab. 1. We conduct an ablation study by replacing the DETR [1], which serves as the object detector in our `RefAtomNet`, by using RetinaNet [5]. Compared with the most outperforming baselines BLIPv2 [3], both `RefAtomNet` (RetinaNet) and `RefAtomNet` (DETR) show promising performance improvements. Regarding the two employed object detectors, there are slight performance changes with $< 1\%$ for each metric,

Table 1: Generalizability to other object detectors

Fusion	mIOU	mAP	AUROC	mIOU	mAP	AUROC
	Val			Test		
BLIPv2 [3]	32.99	52.13	66.56	32.75	53.19	69.92
RefAtomNet (RetinaNet [5])	**38.65**	55.08	69.16	**37.82**	56.67	73.63
RefATomNet (DETR [1])	38.22	**55.98**	**69.73**	36.42	**57.52**	**73.95**

showcasing that our proposed `RefAtomNet` can generalize well to other object detectors. `RefAtomNet` (RetinaNet) can harvest 38.65%, 55.08%, and 69.16% of mIOU, mAP, and AUROC and 37.82%, 56.67%, and 73.63% of mIOU, mAP, and AUROC, on val and test sets, respectively.

E Ablation of the Module Parameters

In this section, we deliver the analysis on the hyperparameters of the proposed `RefAtomNet` by using BLIPv2 [3] as the textual visual backbone and DETR [1] as the object detector to pursue the suitable hyperparameters for the number of heads and the number of agents.

E.1 Ablation of the Frame Number

Overall we get better performance when using 8 frames in training and testing, which is the frame number we used in our main paper.

Table 2: Ablation of the frame number.

Frames	mIOU	mAP	AUROC	mIOU	mAP	AUROC
	Val			Test		
4	37.82	53.21	67.27	34.75	55.00	72.23
6	37.28	53.55	67.50	36.01	55.24	72.14
8	**38.22**	**55.98**	**69.73**	**36.42**	**57.52**	**73.95**
10	37.51	53.41	67.33	36.12	54.87	71.86
12	36.85	53.37	67.36	36.22	55.20	71.90

E.2 Ablation of the Head Number

We deliver the ablation of the head number used for acquiring the Query, Key, Value, and Agent in Tab. 3, where head number $N_h \in [1, 2, 3, 4, 16]$. We observe

Table 3: Ablation of the head number when the agent number is set as 4.

Heads	mIOU	mAP	AUROC	mIOU	mAP	AUROC
	Val			Test		
1	38.22	55.98	69.73	36.42	57.52	73.95
2	35.60	54.91	68.94	34.72	55.02	71.71
3	34.58	54.97	71.71	34.58	54.97	71.71
4	37.17	54.70	68.79	35.63	54.88	71.59
16	36.10	54.40	68.59	35.22	54.68	71.40

Table 4: Ablation of Agent Number when the head number is set as 1.

Agents	mIOU	mAP	AUROC	mIOU	mAP	AUROC
	Val			Test		
1	32.63	56.71	69.70	33.68	55.40	73.08
2	37.97	55.68	69.58	36.89	57.81	74.29
3	36.80	55.86	69.82	35.37	57.32	73.77
4	38.22	55.98	69.73	36.42	57.52	73.95
16	35.47	55.52	69.57	33.96	57.91	74.22

that when $N_h = 1$, the `RefAtomNet` achieves the best performance. We thereby use $N_h = 1$ in our `RefAtomNet`. All the experiments are conducted by selecting the number of agents $N_a = 4$.

E.3 Ablation of the Agent Number

We further show the ablation study of the agent number in Tab. 4, where $N_a \in [1, 2, 3, 4, 16]$. We observe that when N_a=4 the `RefAtomNet` achieves the best performance on val set considering the primary evaluation metrics, *i.e.*, mAP and AUROC. We thereby use N_a=4 in our network setting.

F Discussion of the Model Parameters

We compare the baselines and our proposed method on the prioritized performances in terms of mAP for val and test sets, and the amount of the trainable parameters in Tab. 5. Most of the approaches used for the visual language model preserve more than $100M$ trainable parameters compared to the approaches from the atomic action localization group. Compared with the most outperforming baseline BLIPv2 [3], our proposed new modules only result in the increment of $27M$ parameters, while delivering promising mAP improvements by 3.85% and 4.33%, on the val and test sets. Compared with the method of the largest scale in the baselines, *i.e.*, Singularity, our method delivers 13.55% and 15.34% mAP

Table 5: A comparison of the model trainable parameters and the performance. The mAP on the val and test sets are shown.

Dataset	I3D	X3D	MViTv2-B	VideoMAE2-B	Hiera-B	Singularity	AskAnything	MeVTR	Clip4Clip	XClip	BLIPv2	RefAtomNet
$\mathbf{N}_{Parameters}$	25M	3.76M	71M	87M	52M	203M	0.66M	164M	149M	150M	187M	214M
$\mathbf{mAP}_{Val}$	44.04	44.45	42.32	42.02	42.74	42.43	51.42	38.42	39.48	42.46	52.13	55.98
$\mathbf{mAP}_{Test}$	44.64	46.34	42.60	41.87	41.14	42.18	52.25	36.27	37.17	40.82	53.19	57.52

Table 6: Experimental results of the most outperforming baselines and `RefAtomNet` when rain noise and fog noise perturbations are added into the videos in the test phase.

Method	**Test-time rain noise perturbation**						**Test-time fog noise perturbation**					
	mIOU	mAP	AUROC	mIOU	mAP	AUROC	mIOU	mAP	AUROC	mIOU	mAP	AUROC
	Val			Test			Val			Test		
Singularity [2]	14.44	39.32	56.17	15.35	39.95	53.28	14.05	39.46	56.68	15.84	40.58	53.43
XCLIP [6]	36.92	41.85	55.53	33.10	38.93	57.15	37.24	40.44	52.90	33.99	36.33	52.41
AskAnything [4]	20.63	51.05	65.69	21.91	51.69	68.54	24.37	49.56	63.51	25.27	50.35	66.78
BLIPv2 [3]	31.53	51.26	65.60	31.93	53.15	69.55	35.85	48.53	63.18	35.65	52.55	68.00
Ours	**37.24**	**55.20**	**68.89**	**36.02**	**56.99**	**73.72**	**41.00**	**51.08**	**65.25**	**39.38**	**55.07**	**71.42**

benefits on the val and test sets with only $11M$ more parameters, indicating the effectiveness of our method for RAVAR by suppressing the irrelevant information in the visual stream.

G Robustness against Disturbances on Video during Test Time

During practical usage, the input video has the possibility to be disturbed by different video noises. To assess the robustness of the proposed model and the most outperforming baselines, we conduct a robustness ablation study in Tab. 6 to simulate the rain and fog noises and Tab. 7 for shot and Gaussian noises on the val and test sets according to several perturbation types derived from [7]. In the following, we will deliver the definition of different test time perturbations and the analysis of each perturbation in detail.

G.1 Rain Noise

The process of injecting rain noise into video frames is engineered to emulate the visual manifestation of precipitation within video sequences. This procedure can be delineated through the following steps:

– We generate synthetic raindrops featuring diverse sizes, intensities, and descent angles. This is accomplished by conceptualizing raindrops as ellipses characterized by variable degrees of transparency and blur to mimic motion. Consequently, we represent raindrops as an aggregation of ellipses.

$$R(x,y) = \left\{ (x_i, y_i, r_i^l, r_i^s, \theta_i, \alpha_i) \mid i = 1, \ldots, N \right\}, \tag{1}$$

where (x_i, y_i) are the coordinates of the i-th raindrop, r_i^l and r_i^s denote the long radius and the short radius, θ_i is the falling angle, and α_i is the transparency.
- Then the generated raindrops are inserted onto each frame of the video. This involves blending the raindrop layer with the original video frames using beta blending, where the final pixel value I' is given by:

$$I' = (1 - \beta_i)I + \beta_i R, \tag{2}$$

I is the original pixel intensity, and R is the raindrop intensity.
- Finlay we apply random motion blur to the raindrops to simulate the falling motion. The extent of the blur corresponds to the speed and angle of the rain, enhancing the realism of the effect.

The parameters for raindrop generation, such as size, intensity, angle, and speed, are chosen and varied randomly to simulate natural rain. Additionally, the ambient lighting and camera effects, like reflections and refractions, can also be considered for a more realistic simulation. The transparency α_i is chosen randomly from $[0.9, 0.8, 0.7, 0.6, 0.5]$. The long radius of the ellipse is chosen as 20 pixels while the short radius of the ellipse is chosen as 1 pixel. The position of the raindrop is randomly chosen among all the positions of one frame. The angle θ_i is chosen randomly in $[-10°, 10°]$. β_i is randomly chosen from $[0, 0.5]$. We choose $N = 83$ for each frame in the perturbed videos.

G.2 Fog Noise

The fog noise is intended to replicate the visual phenomenon of fog, which is distinguished by diminished contrast, decreased saturation, and a progressive white overlay that intensifies with distance. The procedure for simulating fog within video frames can be executed as follows. Given:

- Map size $n = ImageSize$,
- Wibble decay factor $d = 3$,
- Initial step size $s = n$,
- Initial wibble value $w = 100$.

The plasma fractal heightmap H is initialized with dimensions n and starting values. At each iteration:

- **Square step**: For each square in the grid:

$$H_{i+\frac{s}{2},j+\frac{s}{2}} = \frac{H_{i,j} + H_{i+s,j} + H_{i,j+s} + H_{i+s,j+s}}{4} + \Delta w, \tag{3}$$

where Δw is a random value from $[-w, w]$ and s is the current step size.
- **Diamond step**: For each diamond in the grid, we calculate the center value as the mean of the four corner points plus a random value:

$$H_{i,j} = \frac{H_{i-\frac{s}{2},j} + H_{i+\frac{s}{2},j} + H_{i,j-\frac{s}{2}} + H_{i,j+\frac{s}{2}}}{4} + \Delta w. \tag{4}$$

– Update the step size and wibble value:

$$s := \frac{s}{2}, \quad w := \frac{w}{d}. \tag{5}$$

This process repeats until the step size $s \geq 2$.

Fog Effect Application. For a given image sequence, the fog effect is applied based on a severity level which determines constants (C_1, C_2) from a predefined set. The fog layer F is generated by multiplying the plasma fractal with constant C_1:

$$F = C_1 \times PlasmaFractal(wibbledecay = C_2). \tag{6}$$

Then, for each image I in the sequence, the fog is applied as follows:

– Scale the image I to the range $[0, 1]$.
– Trim I to the central region of interest.
– Add the fog layer F to I, ensuring the fog does not exceed the original brightness.
– Apply normalization to maintain image contrast:

$$I_{\text{fog}} = \text{clip}\left(\frac{I \times \max(I)}{\max(I) + C_1}, 0, 1\right) \times 255. \tag{7}$$

The result is the fog-enhanced image sequence. We choose $C_1 = 1.5$ and $C_2 = 2.5$ to simulate the fog effect.

G.3 Analysis of the Rain and Fog Noises

The experimental results by injecting rain and fog noises on the val and test sets are delivered in Tab. 6. We conduct experiments on the four most outperforming baselines selected from our benchmark, *i.e.*, Singularity [2], XCLIP [6], AskAnything [4], BLIPv2 [3], and on our proposed method `RefAtomNet`. We first observe that by injecting two types of noise, all the methods show performance decay, while the fog noise demonstrates more negative influence on the RAVAR performances compared with the rain noise, as fog noise will blur more detailed visual cues. `RefAtomNet` delivers the best performances by 37.24%, 55.20%, 68.89% and 36.02%, 56.99%, 73.72% of mIOU, mAP, and AUROC, on val and test sets respectively under test-time rain noise, while delivering 41.00%, 51.08%, 65.25% and 39.38%, 55.07%, 71.42% of mIOU, mAP, and AUROC, on val and test sets respectively under test-time fog noise.

G.4 Shot Noise

Shot noise, alternatively known as Poisson noise, is characterized by fluctuations that conform to a Poisson distribution. This type of noise is predominantly associated with the quantized nature of electronic charges or photons in optical

Table 7: Experimental results of the most outperforming baselines and `RefAtomNet` when shot noise and Gaussian noise perturbations are added into the videos in the test phase.

Method	Test-time shot noise perturbation						Test-time Gaussian noise perturbation					
	mIOU	mAP	AUROC	mIOU	mAP	AUROC	mIOU	mAP	AUROC	mIOU	mAP	AUROC
	Val			Test			Val			Test		
Singularity [2]	8.77	39.20	55.85	9.26	40.34	53.11	6.28	38.00	54.37	6.15	40.36	53.53
XClip [6]	30.65	40.62	53.79	29.47	37.11	54.64	34.15	40.38	53.63	31.48	37.41	54.88
AskAnything [4]	21.52	48.35	62.96	23.29	49.03	66.26	20.80	47.54	60.88	22.41	47.26	64.54
BLIPv2 [3]	32.12	50.31	64.19	32.39	52.24	68.12	31.75	48.63	62.08	32.22	49.81	66.13
Ours	**37.44**	**52.24**	**66.30**	**36.35**	**54.71**	**70.79**	**36.48**	**50.16**	**63.61**	**34.78**	**52.08**	**68.79**

systems. The injection of shot noise into video frames aims to simulate the intrinsic randomness encountered in real camera sensors, which is a consequence of the quantum properties of light. To incorporate shot noise into a video frame, execute the subsequent steps for each pixel within such frame:

- Denote the original pixel value as I, which represents the mean number of photons (or intensity) detected.
- Generate a new pixel value I', which is a random value drawn from the Poisson distribution with mean I. The new value can be represented as:

$$I' \sim \text{Poisson}(s * I) = \frac{e^{-(s*I)}(s * I)^k}{k!}, \tag{8}$$

where k is the actual observed count, and s is the severity chosen as 5.

G.5 Gaussian Noise

Gaussian noise is widely used in video data processing because it closely mimics the natural noise presented in electronic devices and sensors due to thermal motion and other factors. Additionally, its mathematical properties and ease of implementation make it a standard choice for benchmarking and testing video processing algorithms. To inject Gaussian noise into video frames, we follow the following steps for each pixel in each frame:

- We first determine the desired noise level, which is typically characterized by the standard deviation σ of the Gaussian distribution. The mean μ of the distribution is often set to 0 for noise injection purposes.
- For each pixel in the frame with the original intensity value I, we generate a random value n from a Gaussian distribution with mean μ and standard deviation σ. This procedure can be represented as:

$$n \sim \mathcal{N}(\mu, \sigma^2). \tag{9}$$

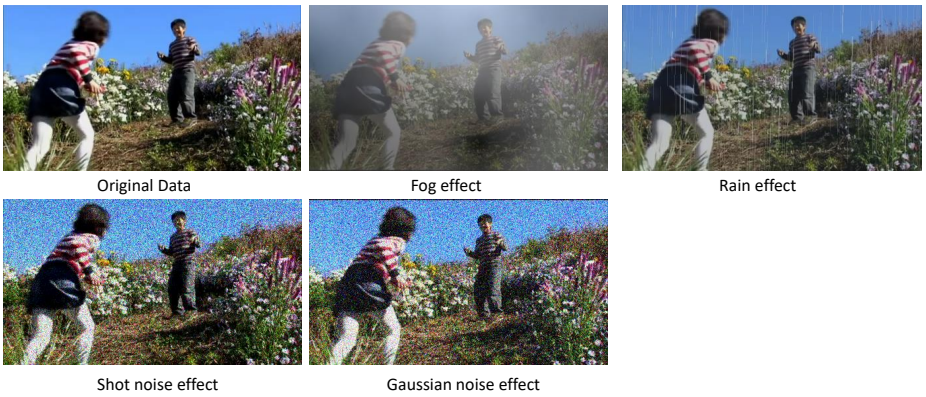

Fig. 1: An overview of a perturbed frame by using four different kinds of perturbation.

– Finally, we add the noise value n to the original pixel value I to obtain the new noisy pixel value I':

$$I' = I + n. \tag{10}$$

The resulting noisy pixel values I' might exceed the valid range of pixel values (*e.g.*, 0 to 255 for an 8-bit image). Therefore, it is common to clip the values to remain within this valid range after the noise has been added. We choose $\mu = 0$ and $\sigma = 0.2$ in our experiments.

G.6 Analysis of the Shot and Gaussian Noises

The experimental outcomes resulted from the application of shot and Gaussian noises on the val and test sets are revealed in Tab. 7. Experiments were conducted with the four highest-performing baselines identified from our benchmarks, namely, Singularity [2], XCLIP [6], AskAnything [4], BLIPv2 [3], as well as our method `RefAtomNet`. Generally, Gaussian noise was deemed to pose a greater challenge than shot noise due to its propensity to obscure finer visual details. Reflected in the results, the introduction of both noise types precipitates a decline in the performance across all evaluated methods, with Gaussian noise exerting a more detrimental impact on the RAVAR performances in comparison to shot noise. In scenarios characterized by test-time shot noise, `RefAtomNet` won the competition, achieving mIOU, mAP, and AUROC scores of 37.44%, 52.24%, 66.30% on the val set, and 36.35%, 54.71%, 70.79% on the test set, respectively. Notably, under conditions of test-time Gaussian noise, `RefAtomNet` reported mIOU, mAP, and AUROC scores of 36.48%, 50.16%, 63.61% on the val set, and 34.78%, 52.08%, 68.79% on the test set, respectively. We further provide the visualizations of these four different kinds of perturbations in Fig. 1.

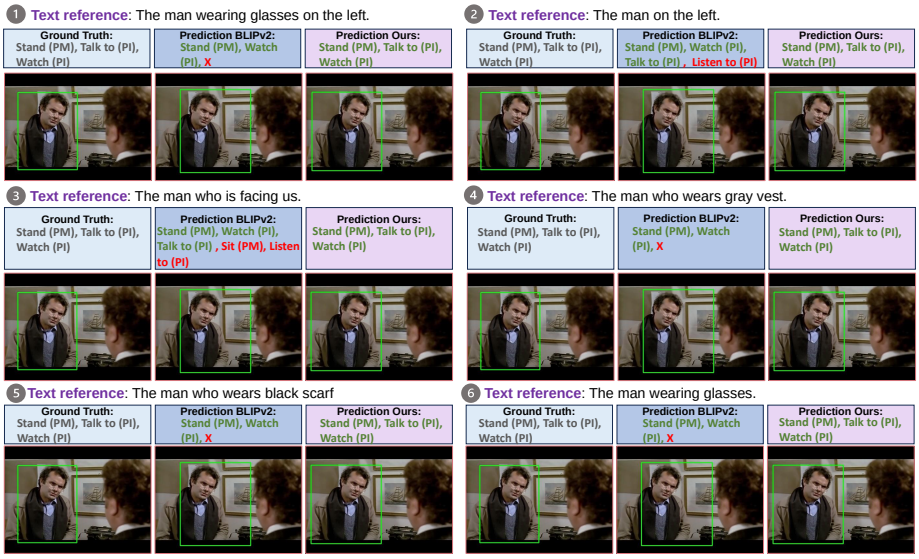

Fig. 2: Qualitative results for the test time rephrasing.

H Qualitative Results for Test Time Rephrasing

We further deliver a sample towards test rephrasing, where we referred the person of interest with different descriptions in Fig. 2. We set the threshold as 0.91 for both of the models to get the multi-label predictions. The person of interest is textually referred to as *the man wearing glasses on the left*, *the man on the left*, *the man who is facing us*, *the man who wears gray vest*, *the man who wears black scarf*, and *the man wearing glasses*, respectively. We observe that the predicted locations of the person do not change among different textual descriptions, varying from the visual appearance attributes leveraged for the indication. The atomic action recognition results of our `RefAtomNet` preserve consistency and deliver concrete predictions for the person of interest. However, there are small fluctuations in the atomic action predictions of the baseline BLIPv2 [3]. These results demonstrate the strong generalizability of our approach towards the varied descriptions during the test time, which is essential for real-world applications since the textual descriptions may differ among different users according to the person's appearance attributes.

I More Samples of the RefAVA Dataset

In this section, we deliver more samples from the contributed RefAVA dataset, as shown in Fig. 3, where for each instance, the textual reference sentence is displayed in the light orange box on the right side of the image, the atomic action annotations and the bounding box are shown on the top of the image and within the image itself using a green box.

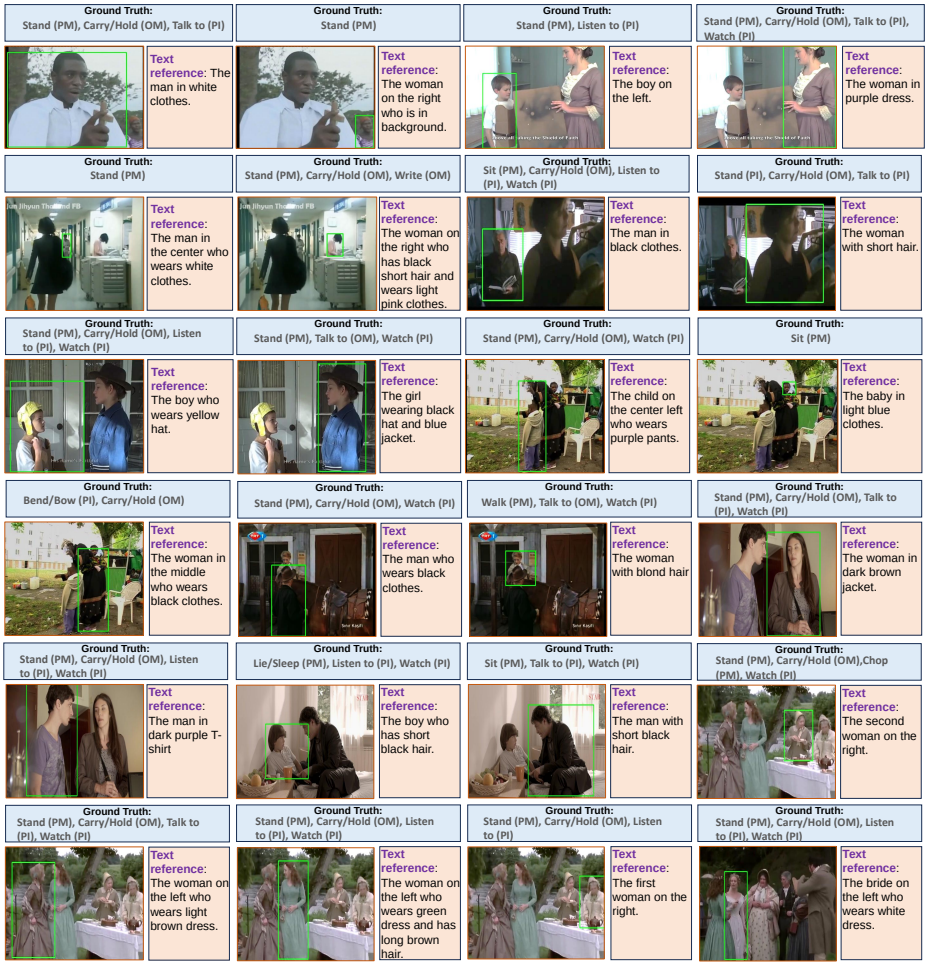

Fig. 3: More samples from our RefAVA dataset.

References

1. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (2020)
2. Lei, J., Li, L., Zhou, L., Gan, Z., Berg, T.L., Bansal, M., Liu, J.: Less is more: ClipBERT for video-and-language learning via sparse sampling. In: CVPR (2021)
3. Li, J., Li, D., Savarese, S., Hoi, S.: BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: ICML (2023)
4. Li, K., He, Y., Wang, Y., Li, Y., Wang, W., Luo, P., Wang, Y., Wang, L., Qiao, Y.: VideoChat: Chat-centric video understanding. arXiv preprint arXiv:2305.06355 (2023)
5. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV (2017)
6. Ma, Y., Xu, G., Sun, X., Yan, M., Zhang, J., Ji, R.: X-CLIP: End-to-end multi-grained contrastive learning for video-text retrieval. In: MM (2022)
7. Yi, C., Yang, S., Li, H., Tan, Y.P., Kot, A.C.: Benchmarking the robustness of spatial-temporal models against corruptions. In: NeurIPS (2021)