Supplementary Material of Abductive Ego-View Accident Video Understanding for Safe Driving Perception

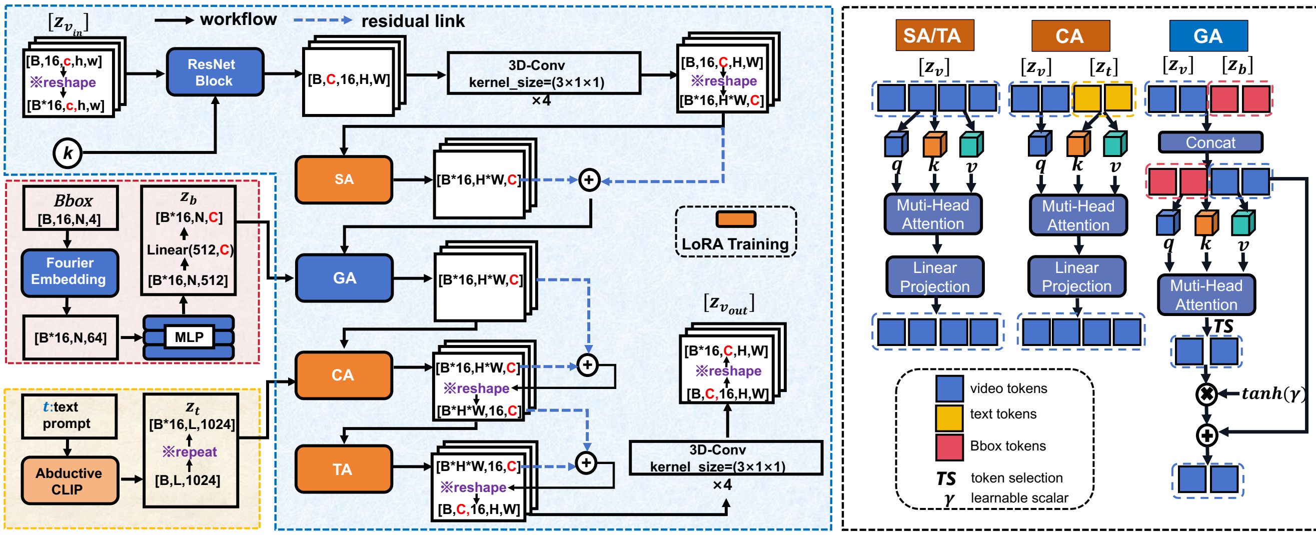

Figure 1. **The detailed workflow of 3D-CAB** in OAVD. 3D-CAB is one layer of the 3D U-net ϕ in Fig. 5 of the main paper body. To be clear, we denote the input representation of 3D-CAB as $\mathbf{z}_{v_{in}}$, and the output representation as $\mathbf{z}_{v_{out}}$. Within 3D-CAB, the feature representation of bounding boxes $\mathbf{z}_b$ and text descriptions $\mathbf{z}_t$ are fused successively to the Gated Self-Attention (GA) and Cross-Attention (CA) modules, where $\mathbf{z}_b$ is obtained by MLP(Fourier($Bbox$)) in Eq.(4) and $\mathbf{z}_t$ is generated by our Abductive CLIP. In Fourier(Bbox), there is a Token Selection (TS) module [34] to find the important tokens for object representation learning. Notably, different from [61], the query ($\mathbf{q}$), key ($\mathbf{k}$), and value ($\mathbf{v}$) are all updated in the OAVD training phase.

1. The Architecture of 3D-CAB

To be clear for re-reproduction, we detail the workflow of 3D-CAB in OAVD, as shown in Fig. 1. B denotes the batch size, and the maximum text prompt length L is set to 77. In each layer of 3D-CAB, c, h, and w represent the channels, height, and width of the input video feature $\mathbf{z}_{v_{in}}$, and C, H, and W represent the channels, height, and width of the video clip representation after ResNet Block encoding. Notably, the channels, height, and width in each step of 3D-CAB change for a dimension adaptation. Furthermore, we inject different attention modules, *i.e.*, **SA**, **CA**, and **TA**, into Low-rank Adaptation (LoRA) trainer [6] for fast fine-tuning on LDM [46].

2. OD Analysis for Different Kinds of Objects

For an adequate benchmark, we offer a more detailed Object Detection (OD) analysis for distinct object types. Likewise, our evaluation utilizes the Average Precision (AP) metrics. In this context, we consider the original AP (average precision with IoU thresholds ranging from 0.5 to 0.95), AP50 (with an IoU threshold of 0.5), and AP75 (with an IoU threshold of 0.75) for our assessment. Additionally, due to the varying scales of the objects involved in collisions during accident scenarios, we have evaluated the proficiency of the model in detecting objects of small ($< 32*32$), medium ($> 32*32$ & $< 96*96$), and large ($> 96*96$) scales, as measured by AP_S, AP_M, and AP_L.

We present the fine-grained quantitative object analysis for 11 state-of-the-art detectors in Tab. 1 and Tab. 2. According to the results, we can see that the accuracy of both detectors, YOLOv5s and DiffusionDet are the best in almost all object categories. YOLOv5s is better than DiffusionDet with V1-Train [, ,] for testing [, ,], while DiffusionDet benefits from excellent generalization (V2-Train [,], test.[]), which allows DiffusionDet to detect important objects in accident scenarios even if these objects are not present in the training data.

Sensitivity to Different Kinds of Objects: According to the results of Tab. 1 and Tab. 2, all object detectors perform best when detecting cars as they are the most commonly occurring object in MM-AU. YOLOv5s obtains 0.936 of AP50 in the V1-Train mode, and DiffusionDet generates 0.908 of AP50 under the V2-Train mode. For cars, pedestrians, trucks, buses, and traffic lights, the **AP** values of the best detector are larger than 0.5. Yet, motorcycles and cyclists are hard to be detected especially under the V2-Train

[6] https://github.com/cloneofsimo/lora

Table 1. The object detection results of V1-Train [, ,]) and V2-Train [,]) for 11 state-of-the-art detectors on the MM-AU, *w.r.t.*, **pedestrians**, **cars**, **motorcycles**, and **trucks**.

pedestrian

	V1-Train [, ,], test. [, ,]						V2-Train [,], test. []					
method	AP	AP50	mAP75	AP_S	AP_M	AP_L	AP	AP50	AP75	AP_S	AP_M	AP_L
FasterRCNN [45]	0.454	0.715	0.522	0.491	0.473	0.376	0.294	0.535	0.302	0.33	0.340	0.149
CornerNet [30]	0.378	0.549	0.439	0.317	0.436	0.252	0.335	0.511	0.384	0.203	0.406	0.217
CascadeRPN [58]	0.448	0.699	0.513	0.443	0.46	0.424	0.365	0.593	0.407	0.331	0.405	0.266
CenterNet [10]	0.011	0.040	0.002	0.037	0.014	0.011	0.047	0.135	0.019	0.034	0.062	0.021
DETR [4]	0.099	0.294	0.038	0.096	0.088	0.15	0.058	0.175	0.022	0.029	0.064	0.067
EfficientNet [53]	0.114	0.299	0.055	0.096	0.127	0.106	0.000	0.002	0.000	0.004	0.000	0.001
Deformable-DeTR [74]	0.404	0.686	0.441	0.396	0.421	0.361	0.369	0.64	0.404	0.317	0.414	0.279
YOLOx [16]	0.424	0.695	0.471	0.387	0.440	0.406	0.293	0.531	0.297	0.206	0.344	0.213
YOLOv5s [23]	**0.529**	**0.784**	**0.632**	0.459	**0.544**	**0.521**	0.370	0.632	0.412	0.295	0.419	0.265
DiffusionDet [7]	0.527	0.767	0.607	**0.463**	**0.544**	0.516	**0.480**	**0.699**	**0.557**	**0.423**	**0.531**	**0.376**
YOLOv8 [54]	0.506	0.748	0.590	0.455	0.516	0.512	0.415	0.650	0.481	0.322	0.463	0.337

car

	V1-Train [, ,], test. [, ,]						V2-Train [,], test. []					
Detectors	AP	AP50	AP75	AP_S	AP_M	AP_L	AP	AP50	AP75	AP_S	AP_M	AP_L
FasterRCNN [45]	0.677	0.910	0.788	0.532	0.672	0.771	0.608	0.851	0.694	0.501	0.629	0.639
CornerNet [30]	0.493	0.628	0.537	0.259	0.561	0.532	0.481	0.639	0.522	0.259	0.563	0.479
CascadeRPN [58]	0.714	0.908	0.805	0.567	0.701	0.819	0.644	0.866	0.733	0.531	0.646	0.706
CenterNet [10]	0.073	0.135	0.071	0.100	0.094	0.062	0.264	0.515	0.242	0.194	0.328	0.256
DETR [4]	0.402	0.746	0.381	0.133	0.349	0.638	0.346	0.676	0.312	0.135	0.308	0.524
EfficientNet [53]	0.409	0.745	0.426	0.140	0.423	0.547	0.146	0.359	0.086	0.050	0.151	0.191
Deformable-DeTR [74]	0.657	0.906	0.763	0.466	0.636	0.801	0.607	0.882	0.684	0.393	0.599	0.736
YOLOx [16]	0.713	0.913	0.799	0.529	0.706	0.840	0.619	0.844	0.692	0.431	0.622	0.720
YOLOv5s [23]	**0.769**	**0.936**	**0.862**	0.585	**0.762**	**0.882**	0.682	0.902	0.787	0.495	0.684	0.773
DiffusionDet [7]	0.754	0.932	0.836	**0.586**	0.747	0.867	**0.720**	**0.908**	**0.801**	**0.575**	**0.721**	**0.808**
YOLOv8 [54]	0.755	0.926	0.836	0.576	0.748	0.867	0.707	0.896	0.791	0.532	0.706	0.801

motorcycle

	V1-Train [, ,], test. [, ,]						V2-Train [,], test. []					
Detectors	AP	AP50	AP75	AP_S	AP_M	AP_L	AP	AP50	AP75	AP_S	AP_M	AP_L
FasterRCNN [45]	0.316	0.554	0.330	0.268	0.341	0.291	0.165	0.342	0.139	0.208	0.200	0.081
CornerNet [30]	0.232	0.393	0.250	0.200	0.284	0.147	0.176	0.334	0.175	0.160	0.222	0.108
CascadeRPN [58]	0.320	0.511	0.340	0.272	0.336	0.313	0.175	0.357	0.150	0.186	0.200	0.0153
CenterNet [10]	0.002	0.008	0.001	0.021	0.003	0.001	0.016	0.052	0.005	0.053	0.019	0.005
DETR [4]	0.115	0.306	0.059	0.057	0.123	0.128	0.038	0.121	0.010	0.029	0.044	0.035
EfficientNet [53]	0.133	0.312	0.085	0.074	0.151	0.127	0.002	0.006	0.000	0.014	0.002	0.001
Deformable-DeTR [74]	0.276	0.506	0.276	0.231	0.305	0.266	0.201	0.417	0.173	0.115	0.223	0.176
YOLOx [16]	0.332	0.560	0.356	0.253	0.365	0.312	0.148	0.318	0.120	0.183	0.189	0.125
YOLOv5s [23]	**0.388**	**0.615**	**0.429**	**0.301**	**0.406**	**0.391**	0.061	0.146	0.040	0.017	0.044	0.105
DiffusionDet [7]	0.375	0.599	0.403	0.300	0.398	0.365	**0.286**	**0.493**	**0.297**	**0.256**	**0.325**	**0.219**
YOLOv8 [54]	0.370	0.578	0.412	0.296	0.390	0.368	0.241	0.440	0.237	0.241	0.271	0.215

truck

	V1-Train [, ,], test. [, ,]						V2-Train [,], test. []					
Detectors	AP	AP50	AP75	AP_S	AP_M	AP_L	AP	AP50	AP75	AP_S	AP_M	AP_L
FasterRCNN [45]	0.505	0.715	0.594	0.389	0.467	0.539	0.338	0.516	0.390	0.286	0.384	0.314
CornerNet [30]	0.410	0.521	0.439	0.203	0.473	0.390	0.398	0.517	0.422	0.181	0.419	0.404
CascadeRPN [58]	0.545	0.715	0.620	0.385	0.493	0.591	0.412	0.574	0.471	0.316	0.379	0.441
CenterNet [10]	0.021	0.040	0.021	0.017	0.018	0.036	0.076	0.161	0.060	0.048	0.102	0.076
DETR [4]	0.287	0.506	0.292	0.098	0.201	0.373	0.18	0.341	0.173	0.053	0.129	0.220
EfficientNet [53]	0.201	0.345	0.225	0.119	0.193	0.218	0.015	0.045	0.004	0.005	0.016	0.014
Deformable-DeTR [74]	0.550	0.741	0.645	0.362	0.476	0.612	0.463	0.649	0.538	0.266	0.42	0.509
YOLOx [16]	0.332	0.560	0.356	0.253	0.365	0.312	0.410	0.595	0.462	0.253	0.371	0.449
YOLOv5s [23]	0.388	0.615	0.429	0.301	0.406	0.391	0.510	0.686	0.600	0.285	0.418	0.575
DiffusionDet [7]	**0.652**	**0.792**	**0.708**	**0.488**	**0.580**	**0.714**	0.549	0.681	0.599	**0.405**	**0.510**	0.582
YOLOv8 [54]	0.370	0.578	0.412	0.296	0.390	0.368	**0.556**	**0.692**	**0.615**	0.344	0.470	**0.615**

Table 2. The object detection results of V1-Train [, ,]) and V2-Train [,]) for 11 state-of-the-art detectors on the MM-AU, *w.r.t.*, **buses**, **traffic lights**, and **cyclists**.

bus

	V1-Train [, ,], test. [, ,]						V2-Train [,], test. [ ]					
Detectors	AP	AP50	AP75	AP_S	AP_M	AP_L	AP	AP50	AP75	AP_S	AP_M	AP_L
FasterRCNN [45]	0.521	0.690	0.615	0.304	0.431	0.580	0.312	0.455	0.356	0.263	0.298	0.328
CornerNet [30]	0.380	0.465	0.408	0.174	0.404	0.376	0.412	0.507	0.443	0.154	0.359	0.461
CascadeRPN [58]	0.522	0.658	0.604	0.263	0.449	0.579	0.395	0.529	0.464	0.214	0.342	0.441
CenterNet [10]	0.003	0.005	0.003	0.001	0.002	0.003	0.027	0.052	0.025	0.028	0.036	0.026
DETR [4]	0.201	0.321	0.219	0.042	0.118	0.258	0.131	0.212	0.141	0.001	0.076	0.167
EfficientNet [53]	0.106	0.169	0.123	0.028	0.108	0.109	0.003	0.008	0.001	0.001	0.002	0.003
Deformable-DeTR [74]	0.511	0.670	0.603	0.266	0.401	0.591	0.484	0.625	0.575	0.282	0.396	0.541
YOLOx [16]	0.595	0.730	0.678	0.336	0.479	0.670	0.417	0.556	0.483	0.136	0.33	0.475
YOLOv5s [23]	**0.685**	**0.794**	**0.757**	**0.400**	**0.541**	**0.767**	0.418	0.553	0.503	0.006	0.238	0.541
DiffusionDet [7]	0.650	0.759	0.707	0.360	0.531	0.721	**0.574**	**0.674**	**0.632**	**0.315**	**0.492**	**0.631**
YOLOv8 [54]	0.668	0.779	0.734	0.371	0.526	0.753	0.533	0.637	0.592	0.123	0.409	0.616

traffic light

	V1-Train [, ,], test. [, ,]						V2-Train [,], test. [ ]					
Detectors	AP	AP50	AP75	AP_S	AP_M	AP_L	AP	AP50	AP75	AP_S	AP_M	AP_L
FasterRCNN [45]	0.487	0.689	0.583	0.434	0.515	0.208	0.371	0.528	0.451	0.325	0.402	0.039
CornerNet [30]	0.412	0.543	0.482	0.306	0.506	0.024	0.248	0.317	0.280	0.275	0.286	0.018
CascadeRPN [58]	0.495	0.675	0.585	0.417	0.532	0.226	0.409	0.531	0.480	0.368	0.437	0.103
CenterNet [10]	0.061	0.127	0.048	0.040	0.085	0.000	0.076	0.167	0.057	0.070	0.094	0.000
DETR [4]	0.132	0.359	0.069	0.062	0.163	0.068	0.079	0.243	0.024	0.048	0.095	0.046
EfficientNet [53]	0.164	0.260	0.169	0.090	0.207	0.011	0.024	0.089	0.000	0.006	0.033	0.000
Deformable-DeTR [74]	0.394	0.669	0.450	0.345	0.420	0.304	0.320	0.585	0.323	0.264	0.35	0.155
YOLOx [16]	0.480	0.667	0.570	0.384	0.546	0.328	0.310	0.458	0.359	0.248	0.346	0.151
YOLOv5s [23]	**0.542**	**0.743**	**0.653**	0.428	**0.590**	0.423	0.356	0.548	0.420	0.297	0.391	0.207
DiffusionDet [7]	0.522	0.703	0.605	**0.440**	0.570	0.344	**0.511**	**0.680**	**0.589**	**0.441**	**0.559**	**0.248**
YOLOv8 [54]	0.526	0.703	0.622	0.413	0.575	**0.429**	0.417	0.570	0.492	0.317	0.465	0.247

cyclist

	V1-Train [, ,], test. [, ,]						V2-Train [,], test. [ ]					
Detectors	AP	AP50	AP75	AP_S	AP_M	AP_L	AP	AP50	AP75	AP_S	AP_M	AP_L
FasterRCNN [45]	0.218	0.391	0.246	0.015	0.248	0.196	0.122	0.255	0.105	0.086	0.144	0.072
CornerNet [30]	0.179	0.297	0.184	0.020	0.196	0.191	0.227	0.370	0.242	0.034	0.257	0.194
CascadeRPN [58]	0.255	0.446	0.276	0.017	0.253	0.318	0.150	0.275	0.166	0.066	0.159	0.171
CenterNet [10]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000
DETR [4]	0.035	0.106	0.011	0.012	0.033	0.051	0.003	0.012	0.002	0.000	0.003	0.007
EfficientNet [53]	0.016	0.039	0.010	0.002	0.018	0.019	0.000	0.000	0.000	0.000	0.000	0.000
Deformable-DeTR [74]	0.231	0.446	0.221	0.038	0.244	0.255	0.166	0.313	0.158	0.088	0.177	0.161
YOLOx [16]	0.235	0.439	0.223	0.048	0.228	0.309	0.137	0.271	0.121	0.081	0.174	0.108
YOLOv5s [23]	**0.368**	**0.601**	**0.423**	**0.058**	0.362	**0.477**	0.005	0.012	0.004	0.000	0.006	0.004
DiffusionDet [7]	0.360	0.577	0.391	0.036	**0.373**	0.418	**0.302**	**0.474**	**0.319**	**0.095**	**0.318**	**0.291**
YOLOv8 [54]	0.314	0.508	0.351	0.057	0.302	0.416	0.198	0.332	0.211	0.070	0.236	0.221

mode, where all kinds of AP values are less than 0.5. Here, compared with DiffusionDet, YOLOv5s is with failure on motorcycles and cyclists in the V2-Train mode.

Adaptability to Small Objects: Small object detection is a difficult problem because there are not enough details to obtain a strong feature representation. As for accident scenarios, this problem may be aggravated because of the unusual property. Therefore, we can observe that most detectors generate the lowest AP_S values within their AP value set. For motorcycles, traffic lights, and pedestrians, too large objects commonly are unusual and AP_L values are the smallest in V2-Train mode. Contrarily, for these kinds of objects, AP_L values in V1-Train mode are not the smallest, which indicates that the large size of objects in the accident window frequently appears due to the severe scale change, *e.g.*, the ego-car involved cases in Fig. 2 (1)-(2) and (5)-(6).

Scalability to Corner Objects: The objects in the road accident window are the typical corner cases in object detection. Fig. 2 demonstrates some examples of the detection results of CenterNet, DETR, DiffusionDet, and YOLOv5s. It is clear that these corner cases are hard to address because of the dramatic scale change (Fig. 2(1)-(2) and (5)-(6)) and severe pose distortion (Fig. 2(1) and (3)-(4)). Many objects are wrongly detected, such as the wrong detections of

Figure 2. The object detection snapshots in accident frames by CenterNet [30], DETR [4], DiffusionDet [7], and YOLOv5s [23]. We can see that all detectors fail to detect the cyclist (column (2)) and the pedestrian with distorted posture (column (1)). DETR is more active for covering all possible objects while many false detections are generated.

car→*truck*, *bus*→*truck*. DETR is more active in covering all possible objects while generates many false detections.

In summary, due to the corner cases, object detection in ego-view accident videos still has many unresolved issues.

3. ArA Case Analysis, *w.r.t.*, Different Objects

Continuing the aforementioned analysis of the ArA task in the main body, we show some cases with respect to different objects in Fig. 3 from the results of the state-of-the-art methods. We can see that because many pedestrian-involved accidents may be caused by distracted walking or aggressive movement, such as sudden crossing, besides HCRN [31], all the methods can provide an accurate accident reason for the shown cases. For the surrounding car-involved cases, the irregular behaviors of cars are the common reason for the accidents, which implies a traffic rule reasoning problem. Therefore, the methods with better commonsense knowledge learning, such as SeViLA [70] (the only method for the accurate ArA for the $4th$ case), have advantages. As for the ego-car involved cases, the severe scale change advocates the object-centric methods with better region context learning.

4. More Evaluations of OAVD

More evaluations are provided here for a sufficient understanding of our Object-centric Accident Video Diffusion (OAVD). We provide more example analysis to check the abductive ability by our OAVD with a comparison to other state-of-the-art video diffusion methods. Notably, we further include ModelScope T2V (preprint)[7] and Text2Video-Zero (published in ICCV2023)[8] in the evaluation. ModelScope T2V is re-trained by a same number of samples with our OAVD (*i.e.*, 6000 Co-CPs), and Text2Video-Zero is another training-free video diffusion method.

More Visualizations of OAVD Against SOTAs: Fig. 4 and Fig. 5 present the qualitative comparisons of different video diffusion models. The inference flow is $(Bboxes \rightarrow V_r)$+t_r/t_p→V_g, *i.e.*, that we input the detected bounding boxes $Bboxes$, the video clip in near-accident window V_r, and the accident reason or prevention advice description t_r/t_p. From the demonstrated snapshots, we can see that, our OAVD similarly shows an "*in advance*" phenomenon for the accident reason prompt and eliminates the crashing object when inputting the prevention advice description. ModelScope T2V also generates promising video frames with clear details, even with the ability to eliminate the objects to be involved in accidents after inputting the prevention advice description, as shown by the second example in Fig. 4 and the first case in Fig. 5. Yet, it is not

[7] https://modelscope.cn/models/damo/text-to-video-synthesis/summary

[8] https://github.com/Picsart-AI-Research/Text2Video-Zero

Figure 3. The **case visualization of Accident reason Answering (ArA)** by 8 state-of-the-art Video Question Answering (VQA) methods.

stable verified by Fig. 4 (the $1st$ example) and Fig. 5 (the $2nd$ sample). As for other methods, including the training-free ones, the style and the content of the generated video frames are not relevant to the given text prompts.

More Analysis on the Impact of Bboxes: To be clear about the impact of bounding boxes (**Bboxes**) for our OAVD model, we re-train the OAVD without the input of Bboxes driven by the Sequential CLIP (S-CLIP) and Ab-

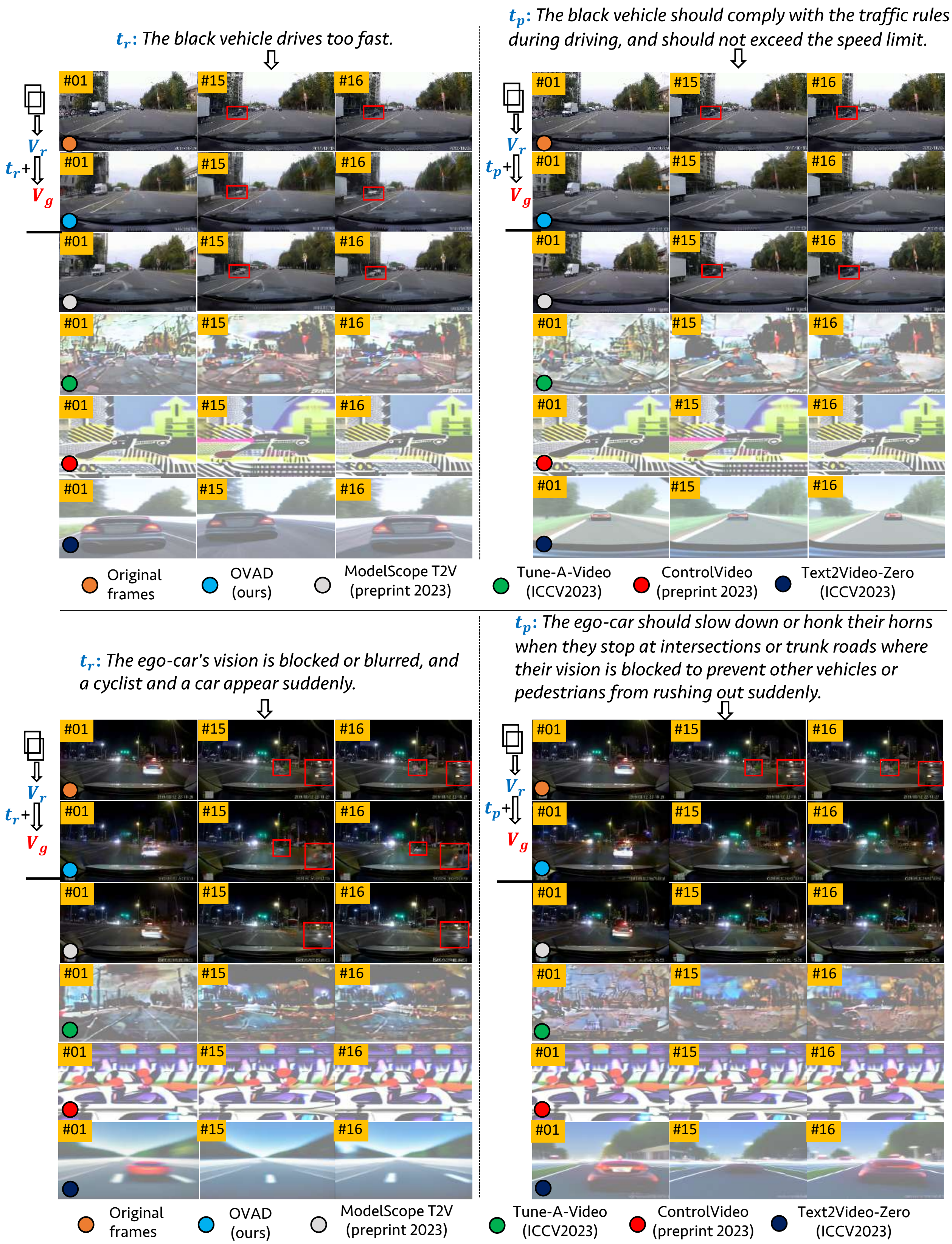

Figure 4. The visualization of generated frames by our OAVD, ModelScope T2V, Tune-A-Video [61], ControlVideo [72], and Text2Video-Zero.

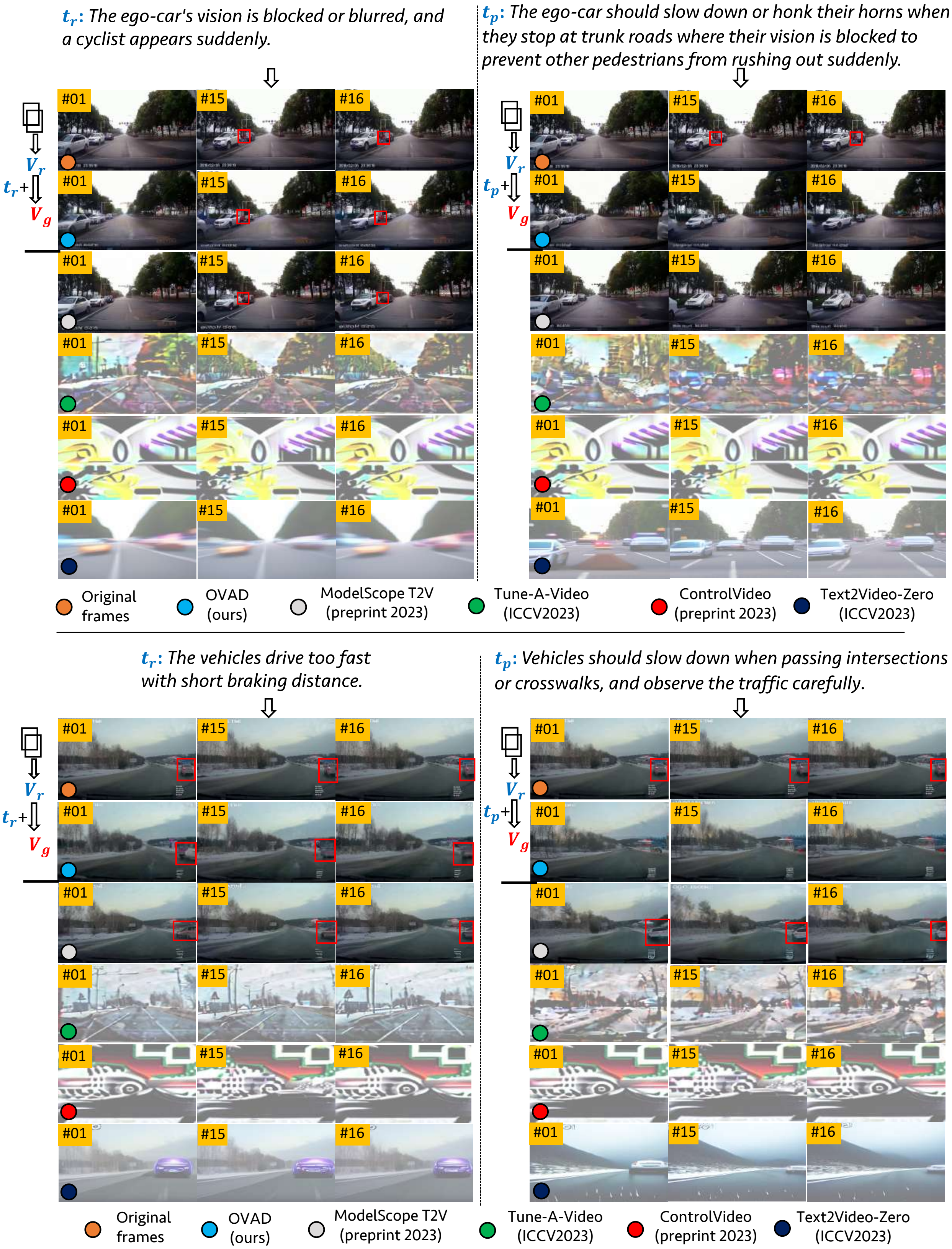

Figure 5. The visualization of generated frames by our OAVD, ModelScope T2V, Tune-A-Video [61], ControlVideo [72], and Text2Video-Zero.

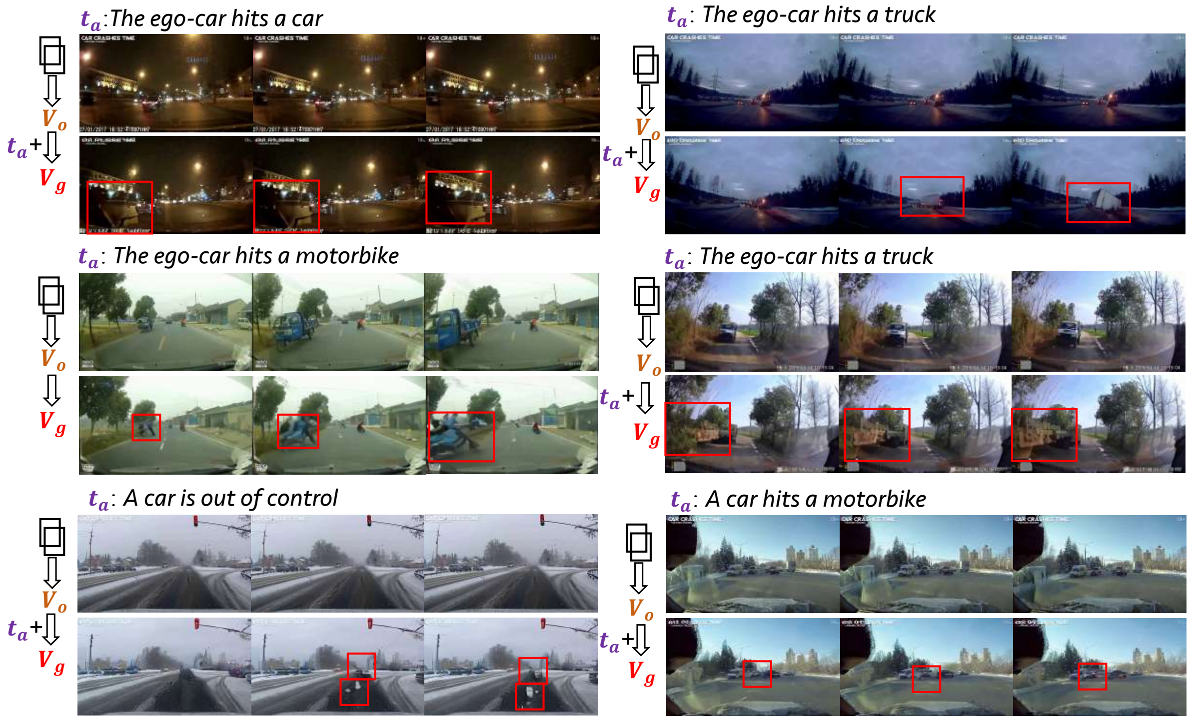

Figure 6. The visualization of **accident video generation** of OAVD with the inference path of $(Bboxes \rightarrow V_o)+t_a \rightarrow V_g$.

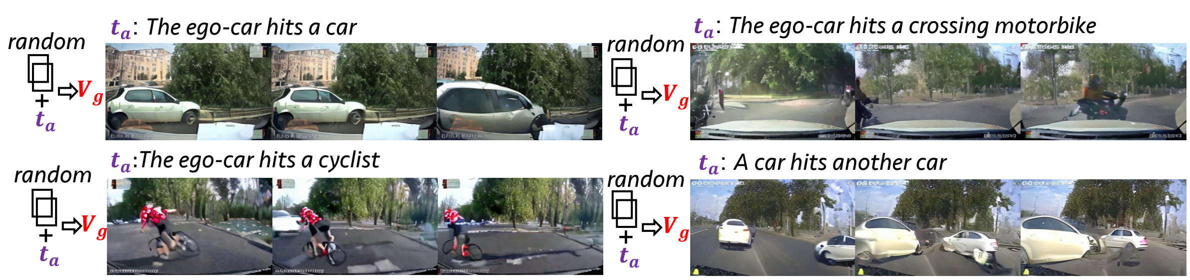

Figure 7. The visualization of **video-free accident video generation** of OAVD with the inference path of $Bboxes+t_a \rightarrow V_g$.

ductive CLIP (A-CLIP) models. The video-level Fréchet Video Distance (**FVD**) [55] is adopted here. The results in Tab. 3 show that the bounding boxes are useful for enhancing the video quality, and lower FVD values are generated. Based on the evaluation, object-centric video diffusion is promising for generating detailed frame content.

Table 3. FVD value comparison of our OAVD with or without the input of bounding boxes. *: with the input of bounding boxes.

Method	OAVD (S-CLIP)*	OAVD (S-CLIP)	OAVD (A-CLIP)*	OAVD (A-CLIP)
FVD ↓	**5372.3**	5384.6	**5238.1**	5358.8

Visualizations of Accident Video Generation: Besides the abductive check for our video diffusion model OAVD, we also show its ability for flexible accident video generation. To be clear, the inference stage here takes the video clip in normal video segment V_o and the accident category description t_a. This configuration verifies the reality-changing ability from normal situations to accidents. Fig. 6 shows some examples of accident video generation. We can curiously find that our OAVD can create the object to be involved in accidents with a clear pose or appearance. This ability may address the few-shot sample issue of accident videos for future task use.

In addition, we also check the video-free accident video generation by only inputting the bounding boxes to our OAVD. Here, the four [9] bounding boxes are randomly set for each example. From the results in Fig. 7, the guidance of the accident category description is clearly verified and the generated accident videos are more realistic without the restriction of original video frames. From these visualizations, OAVD can flexibly augment the video sample scale of ego-view accidents for safe driving.

[9]Other values can also be set.