

R²-Bench: Benchmarking the Robustness of Referring Perception Models under Perturbations

Xiang Li¹, Kai Qiu¹, Jinglu Wang², Xiaohao Xu³, Rita Singh¹, Kashu Yamazaki¹, Hao Chen¹, Xiaonan Huang³, Bhiksha Raj^{1,4}

¹ CMU, ² Microsoft Research Asia, ³ University of Michigan, ⁴ MBZUAI

1 Appendix

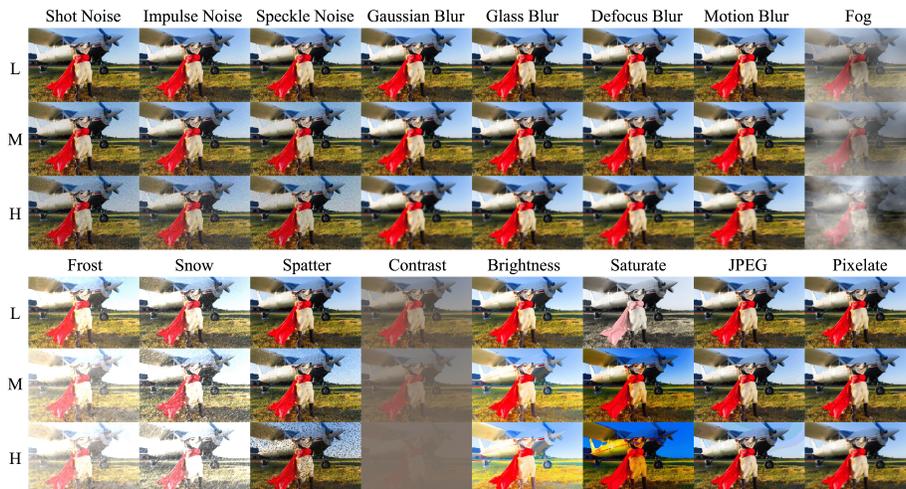


Fig. 1: Visualization of visual perturbations.

1.1 More Visualization of Perturbations

Visual perturbations. Visual perturbations involve the modification of images or video frames to introduce noise, blur, environmental interference, and post-processing in appearance that a system might encounter in real-world scenarios. Examples of visual perturbations can be found in Fig. 1

Acoustic perturbations. Acoustic perturbations alter audio data to test the durability of speech recognition and audio processing systems. We introduce noise, signal adjustments, filtering, physical interference, and post-processing that a system might be affected. Examples of acoustic perturbations are in “acoustic_perturb_example” folder.

Original		AI shapes our future with each innovation.	Innovation fuels progress with each step.
MS	L	AI shapes our future with each innovation.	Innovation fuels progress with each step.
	M	AI shapes our. fur43t with each innovation.	Innovation fyuls pr0gress. with 3each step.
	H	AI sh4pes our f0urture with e4ch innovation 0.	Innovat10n f00ls progress w1th each step.
MP	L	AI sh?apes our future with each innovation.	Innovation fuels p!rogress with each st.ep.
	M	AI shap.es our. future with each. innov.a.tion.	Innovation f.uels progress .wi.th ?each ste:p.
	H	AI sha;pes: our fluture; with each innov;ati.o.n.	I,nnovation fuells pro,gress; w;it,h each! s;tep.
GE	L	AI shape our future with each innovation.	Innovations fuels progress with each steps.
	M	AI shapes our future with each innovation's.	Innovation fuel progress with each steps.
	H	Als shapes our futues with innovation.	Innovate fueling progresses with every steps.
CM	L	AI shapes or future ith each innovation.	Innovtin fuels progress with each step.
	M	Alshapes our futre with eac inoaton.	Inovation fuelsrogrss witeach step.
	H	shpe urfuture wth ech nnovation.	Innvitin fuel prgss wt ech step.

Table 1: Illustration of textual perturbations Misspelling(MS), Misspunctuation(MP), Grammar error(GE) and Character missing(CM) for two example sentence under low(L), medium(M) and high(H) perturbation levels.

Textual perturbations. Textual perturbations involve the modification of text data to introduce errors or variations, akin to those that might caused by human-writing and machine communication in natural language processing. Examples included in Tab. 1

1.2 Performance Evaluation

Referring Image Segmentation. As shown in Tabs. 2 to 10, we test State-of-the-art models' performance in dataset RefCOCO/+ /g [7, 17] in low, medium, and high perturbation levels. After a detailed evaluation, we noticed that PolyFormer [6] stands out with the highest mIoU across all perturbation levels in different datasets, further interpreting our hypothesis on its robustness due to its special polygon representation. SEEM [19], while having lower scores on MIoU and Precision at 50-70, still shows its promising ability to resilience on perturbation, especially in its work on Precision at a high level, which can prove that this model still has a good understanding of the object's overall shape.

Method	L						M						H					
	mIoU	P@50	P@60	P@70	P@80	P@90	mIoU	P@50	P@60	P@70	P@80	P@90	mIoU	P@50	P@60	P@70	P@80	P@90
LAVT [13]	0.58	0.65	0.61	0.55	0.46	0.22	0.51	0.57	0.53	0.48	0.37	0.17	0.44	0.49	0.45	0.39	0.30	0.13
PolyFormer [6]	0.70	0.82	0.78	0.71	0.57	0.21	0.64	0.75	0.71	0.64	0.50	0.18	0.57	0.66	0.62	0.55	0.42	0.14
X-Decoder [18]	0.60	0.69	0.66	0.61	0.51	0.24	0.52	0.61	0.58	0.53	0.44	0.19	0.44	0.51	0.48	0.44	0.35	0.15
ETRIS [12]	0.64	0.75	0.70	0.62	0.46	0.14	0.57	0.66	0.61	0.53	0.38	0.11	0.49	0.56	0.51	0.43	0.29	0.08
SEEM [19]	0.59	0.66	0.64	0.62	0.55	0.34	0.54	0.61	0.60	0.57	0.51	0.30	0.47	0.53	0.51	0.47	0.40	0.22

Table 2: Robustness of referring image segmentation on RefCOCO Validation set with low (L), medium (M) and high (H) perturbation levels.

Video object segmentation. In Tab. 14, We show a detailed evaluation of state-of-the-art methods on the YouTube-VOS [11] dataset under 'Clean', 'Static', and 'Dynamic' mode of perturbations. Among all the evaluated methods, Cutie [2]

Method	L						M						H					
	mIoU	P@50	P@60	P@70	P@80	P@90	mIoU	P@50	P@60	P@70	P@80	P@90	mIoU	P@50	P@60	P@70	P@80	P@90
LAVT [13]	0.64	0.73	0.69	0.63	0.53	0.25	0.56	0.64	0.60	0.54	0.43	0.19	0.48	0.54	0.50	0.44	0.33	0.13
PolyFormer [6]	0.72	0.84	0.81	0.75	0.58	0.19	0.66	0.78	0.75	0.68	0.52	0.17	0.58	0.68	0.65	0.58	0.42	0.13
X-Decoder [18]	0.62	0.71	0.69	0.65	0.53	0.22	0.55	0.65	0.62	0.58	0.47	0.18	0.46	0.54	0.51	0.46	0.36	0.12
ETRIS [12]	0.68	0.80	0.76	0.68	0.51	0.14	0.61	0.71	0.67	0.59	0.43	0.11	0.52	0.61	0.56	0.47	0.31	0.07
SEEM [19]	0.63	0.70	0.69	0.66	0.58	0.34	0.57	0.65	0.63	0.61	0.53	0.28	0.49	0.56	0.54	0.50	0.43	0.22

Table 3: Robustness of referring image segmentation on RefCOCO Test-A set with low (L), medium (M) and high (H) perturbation levels.

Method	L						M						H					
	mIoU	P@50	P@60	P@70	P@80	P@90	mIoU	P@50	P@60	P@70	P@80	P@90	mIoU	P@50	P@60	P@70	P@80	P@90
LAVT [13]	0.53	0.58	0.53	0.47	0.39	0.21	0.47	0.52	0.47	0.41	0.33	0.16	0.40	0.44	0.40	0.34	0.26	0.12
PolyFormer [6]	0.67	0.77	0.73	0.66	0.54	0.25	0.61	0.70	0.66	0.59	0.47	0.20	0.55	0.63	0.58	0.52	0.41	0.17
X-Decoder [18]	0.51	0.58	0.55	0.52	0.44	0.24	0.46	0.52	0.49	0.45	0.37	0.18	0.40	0.45	0.42	0.38	0.31	0.15
ETRIS [12]	0.60	0.68	0.62	0.54	0.41	0.16	0.52	0.59	0.53	0.44	0.32	0.11	0.46	0.51	0.45	0.37	0.26	0.09
SEEM [19]	0.50	0.56	0.54	0.52	0.47	0.31	0.47	0.52	0.50	0.48	0.42	0.26	0.41	0.45	0.43	0.40	0.35	0.22

Table 4: Robustness of referring image segmentation on RefCOCO Test-B set with low (L), medium (M) and high (H) perturbation levels.

Method	L						M						H					
	mIoU	P@50	P@60	P@70	P@80	P@90	mIoU	P@50	P@60	P@70	P@80	P@90	mIoU	P@50	P@60	P@70	P@80	P@90
LAVT [13]	0.58	0.66	0.61	0.56	0.46	0.23	0.51	0.57	0.53	0.48	0.40	0.18	0.43	0.48	0.44	0.39	0.29	0.12
PolyFormer [6]	0.65	0.75	0.72	0.66	0.53	0.20	0.57	0.66	0.63	0.57	0.45	0.16	0.50	0.58	0.55	0.49	0.37	0.12
X-Decoder [18]	0.60	0.69	0.66	0.62	0.52	0.24	0.51	0.59	0.56	0.52	0.42	0.19	0.44	0.51	0.49	0.44	0.35	0.15
ETRIS [12]	0.56	0.65	0.60	0.53	0.40	0.12	0.48	0.55	0.51	0.45	0.32	0.09	0.40	0.46	0.41	0.35	0.23	0.06
SEEM [19]	0.58	0.65	0.63	0.60	0.54	0.34	0.51	0.57	0.55	0.53	0.47	0.28	0.44	0.50	0.48	0.45	0.39	0.21

Table 5: Robustness of referring image segmentation on RefCOCO+ Validation set with low (L), medium (M) and high (H) perturbation levels.

Method	L						M						H					
	mIoU	P@50	P@60	P@70	P@80	P@90	mIoU	P@50	P@60	P@70	P@80	P@90	mIoU	P@50	P@60	P@70	P@80	P@90
LAVT [13]	0.63	0.71	0.68	0.62	0.52	0.25	0.55	0.63	0.59	0.53	0.42	0.18	0.47	0.53	0.49	0.43	0.33	0.14
PolyFormer [6]	0.68	0.79	0.76	0.70	0.56	0.19	0.60	0.71	0.68	0.62	0.48	0.15	0.53	0.62	0.59	0.53	0.40	0.12
X-Decoder [18]	0.61	0.70	0.68	0.64	0.54	0.22	0.54	0.63	0.61	0.57	0.46	0.18	0.44	0.52	0.49	0.44	0.35	0.13
ETRIS [12]	0.62	0.73	0.68	0.61	0.42	0.12	0.53	0.63	0.59	0.52	0.37	0.09	0.44	0.51	0.47	0.40	0.27	0.06
SEEM [19]	0.62	0.70	0.69	0.66	0.59	0.34	0.56	0.63	0.62	0.59	0.53	0.29	0.48	0.54	0.53	0.50	0.43	0.22

Table 6: Robustness of referring image segmentation on RefCOCO+ Test-A set with low (L), medium (M) and high (H) perturbation levels.

Method	L						M						H					
	mIoU	P@50	P@60	P@70	P@80	P@90	mIoU	P@50	P@60	P@70	P@80	P@90	mIoU	P@50	P@60	P@70	P@80	P@90
LAVT [13]	0.52	0.58	0.53	0.47	0.39	0.21	0.46	0.50	0.46	0.42	0.32	0.17	0.39	0.43	0.38	0.33	0.25	0.12
PolyFormer [6]	0.58	0.65	0.61	0.56	0.45	0.21	0.52	0.59	0.55	0.50	0.40	0.18	0.46	0.52	0.48	0.43	0.33	0.14
X-Decoder [18]	0.52	0.59	0.56	0.52	0.45	0.24	0.45	0.51	0.48	0.44	0.37	0.19	0.39	0.44	0.41	0.38	0.31	0.14
ETRIS [12]	0.47	0.52	0.47	0.41	0.29	0.11	0.40	0.44	0.39	0.33	0.24	0.09	0.34	0.37	0.31	0.25	0.17	0.06
SEEM [19]	0.49	0.54	0.52	0.50	0.49	0.29	0.44	0.48	0.47	0.44	0.39	0.25	0.40	0.44	0.42	0.39	0.34	0.20

Table 7: Robustness of referring image segmentation on RefCOCO+ Test-B set with low (L), medium (M) and high (H) perturbation levels.

achieves the highest performance on the Clean dataset. However, it lacks resilience when it comes to perturbations, which can be attributed to its heavy reliance on pixel- and object-level consistency across frames. DEVA [3], leveraging its unique technique of decoupling video on external data and bi-directional propagation,

Method	L						M						H					
	mIoU	P@50	P@60	P@70	P@80	P@90	mIoU	P@50	P@60	P@70	P@80	P@90	mIoU	P@50	P@60	P@70	P@80	P@90
LAVT [13]	0.55	0.60	0.55	0.48	0.38	0.17	0.47	0.51	0.46	0.39	0.30	0.13	0.38	0.42	0.37	0.30	0.21	0.08
PolyFormer [6]	0.64	0.74	0.69	0.61	0.47	0.18	0.57	0.67	0.62	0.54	0.40	0.15	0.50	0.58	0.54	0.46	0.33	0.12
X-Decoder [18]	0.58	0.65	0.62	0.56	0.46	0.22	0.50	0.56	0.53	0.49	0.39	0.18	0.44	0.49	0.46	0.40	0.31	0.13
ETRIS [12]	0.53	0.61	0.55	0.47	0.33	0.11	0.45	0.51	0.46	0.38	0.26	0.08	0.38	0.42	0.36	0.29	0.19	0.06
SEEM [19]	0.56	0.62	0.60	0.56	0.49	0.29	0.49	0.55	0.52	0.49	0.43	0.24	0.43	0.48	0.46	0.42	0.35	0.18

Table 8: Robustness of referring image segmentation on RefCOCOg-umd Validation set with low (L), medium (M) and high (H) perturbation levels.

Method	L						M						H					
	mIoU	P@50	P@60	P@70	P@80	P@90	mIoU	P@50	P@60	P@70	P@80	P@90	mIoU	P@50	P@60	P@70	P@80	P@90
LAVT [13]	0.55	0.61	0.55	0.48	0.38	0.17	0.47	0.52	0.47	0.41	0.31	0.13	0.39	0.42	0.37	0.31	0.23	0.09
PolyFormer [6]	0.64	0.74	0.69	0.61	0.47	0.18	0.58	0.68	0.63	0.56	0.42	0.15	0.50	0.58	0.54	0.47	0.34	0.12
X-Decoder [18]	0.60	0.69	0.65	0.60	0.49	0.23	0.52	0.60	0.57	0.52	0.42	0.18	0.44	0.51	0.48	0.42	0.33	0.15
ETRIS [12]	0.53	0.60	0.55	0.47	0.34	0.11	0.46	0.52	0.46	0.38	0.27	0.08	0.38	0.42	0.36	0.30	0.19	0.05
SEEM [19]	0.57	0.64	0.62	0.58	0.52	0.31	0.51	0.58	0.56	0.52	0.45	0.26	0.45	0.51	0.48	0.45	0.37	0.20

Table 9: Robustness of referring image segmentation on RefCOCOg-umd Test set with low (L), medium (M) and high (H) perturbation levels.

Method	L						M						H					
	mIoU	P@50	P@60	P@70	P@80	P@90	mIoU	P@50	P@60	P@70	P@80	P@90	mIoU	P@50	P@60	P@70	P@80	P@90
LAVT [13]	0.69	0.78	0.74	0.68	0.56	0.27	0.57	0.64	0.60	0.53	0.42	0.18	0.46	0.51	0.46	0.40	0.29	0.12
PolyFormer [6]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
X-Decoder [18]	0.63	0.72	0.69	0.64	0.52	0.25	0.54	0.61	0.59	0.53	0.43	0.19	0.47	0.54	0.50	0.45	0.35	0.15
ETRIS [12]	0.49	0.56	0.50	0.43	0.30	0.09	0.45	0.51	0.45	0.38	0.26	0.08	0.38	0.41	0.37	0.30	0.19	0.05
SEEM [19]	0.61	0.69	0.67	0.63	0.56	0.34	0.53	0.60	0.58	0.54	0.47	0.26	0.46	0.51	0.49	0.45	0.38	0.21

Table 10: Robustness of referring image segmentation on RefCOCOg-google Validation set with low (L), medium (M) and high (H) perturbation levels.

maintains its high performance in 'static' perturbation which somehow preserves the temporal consistency. Notably, DeAOT [15] accomplishes the best result in 'dynamic' perturbation, which can be further thanks to its decoupling of object-agnostic and object-specific features in hierarchical propagation.

Referring video object segmentation. According to the data depicted in Tabs. 11 and 12, we recognize that SgMg [8] exhibits resilience against various types of disturbances, a trait largely attributable to its backbone based on the Swin Transformer, in contrast to other models that utilize a ResNet-based architecture. In the remaining three models, OnlineRefer [9] stands out due to its per-frame prediction mechanism, which provides it with the strongest robustness against perturbations. In Tab. 13, we compare transformer-based model and ResNet-based model separately; OnlineRefer [9] and SgMg [8] also maintain their performance.

1.3 Statistics of Generated Datasets

We show the statistics of generated datasets as shown in Tab. 15

1.4 Implementation of Image Perturbations

Distortions based on Noise.

Method	Low								Medium							
	Anno0		Anno1		Anno2		Anno3		Anno0		Anno1		Anno2		Anno3	
	\mathcal{J}	\mathcal{F}														
ReferFormer [10]	47.9	52.0	47.5	51.4	45.5	49.1	45.2	49.7	43.6	48.4	44.3	49.2	42.6	47.4	41.8	47.1
R ² -VOS [5]	50.3	57.1	49.4	56.6	49.3	56.6	51.7	57.5	47.7	55.2	47.7	55.7	44.3	51.5	46.2	52.6
OnlineRefer [9]	51.4	57.7	50.4	56.8	52.5	58.9	52.2	58.2	48.0	54.2	50.3	56.1	46.9	53.6	46.9	53.5
SgMg [8]	58.2	62.9	57.0	62.9	57.5	62.8	56.5	61.6	51.3	57.3	55.3	61.0	49.9	55.1	51.0	57.2

Method	High								Dynamic							
	Anno0		Anno1		Anno2		Anno3		Anno0		Anno1		Anno2		Anno3	
	\mathcal{J}	\mathcal{F}														
ReferFormer [10]	39.4	43.3	39.0	42.6	37.8	41.0	37.7	40.6	43.0	47.3	44.1	48.0	43.0	47.0	41.4	46.3
R ² -VOS [5]	38.5	43.7	38.5	44.2	39.2	45.3	40.7	45.2	46.8	52.1	46.0	52.2	43.5	50.5	49.6	55.0
OnlineRefer [9]	39.1	43.3	43.4	47.6	42.9	47.2	43.5	47.2	50.5	56.3	49.9	55.2	49.5	54.5	44.3	50.4
SgMg [8]	47.8	51.5	48.9	51.9	46.2	50.6	46.1	50.3	53.3	58.5	53.7	58.5	54.9	60.2	51.4	56.3

Table 11: Robustness of referring video object segmentation methods under low (L), medium (M), and high (H) perturbation levels. Methods in black use RestNet backbone and the method in gray is Transformer based.

Method	Environment								Sensor							
	Snow		Fog		Frost		Spatter		Bright		Defocus		Gau. B		Motion	
ReferFormer [10]	42.8	46.2	45.9	50.3	41.4	45.3	45.9	49.5	48.2	53.0	41.8	45.3	42.8	46.3	41.0	44.8
R ² -VOS [5]	44.3	49.4	48.2	54.1	36.7	42.6	48.0	54.7	53.8	59.6	45.8	50.1	47.6	52.2	45.7	50.3
OnlineRefer [9]	46.0	50.0	48.6	54.1	40.0	45.1	50.6	56.3	54.4	60.8	48.1	53.4	50.1	55.2	46.8	51.2
SgMg [8]	52.8	56.4	54.7	59.2	48.7	53.2	55.4	59.9	58.1	63.4	53.4	57.7	55.1	59.5	52.0	57.4

Method	Sensor								Transmission							
	Glass		Impulse		Shot		Speckle		Contrast		Saturate		JPEG		Pixelate	
ReferFormer [10]	39.6	42.5	42.7	47.2	41.1	45.4	43.4	47.2	45.3	49.5	49.0	53.1	48.0	52.1	43.6	47.5
R ² -VOS [5]	43.1	48.2	41.3	48.5	40.5	46.6	41.2	47.3	46.7	52.2	54.5	60.3	52.6	58.6	48.8	54.2
OnlineRefer [9]	41.9	46.2	47.5	52.6	46.6	51.6	47.3	52.6	46.6	51.1	55.1	61.8	53.1	59.8	48.5	55.2
SgMg [8]	49.8	54.7	51.3	55.3	51.1	54.7	51.8	55.8	54.2	58.4	57.8	62.8	57.4	62.3	54.3	59.2

Table 12: Robustness of referring video object segmentation methods under different textual and visual perturbations on AVS-Multi. Methods in black use RestNet backbone and the method in gray is Transformer based.

1. **Gaussian Noise.** For an original image denoted as I , the addition of Gaussian noise introduces a degraded version I' , described by

$$I' = I + \eta \quad (1)$$

where η , the noise, follows a Gaussian distribution with a mean of zero and a variance of σ^2 , depicted as $\eta \sim \mathcal{N}(0, \sigma^2)$.

2. **Shot Noise.** The phenomenon of shot noise arises from the discrete nature of photon or particle detection in imaging, which can be modeled by a Poisson distribution.
3. **Impulse Noise.** In this scenario, select pixels in the original image I are altered to extreme limits (*e.g.*, maximum or minimum RGB values) based on a probability p . The alteration process for a pixel located at (x, y) is defined as

$$I'(x, y) = \begin{cases} \text{minimum value} & \text{with probability } p/2 \\ \text{maximum value} & \text{with probability } p/2 \\ I(x, y) & \text{otherwise} \end{cases} \quad (2)$$

Table 13: Robustness of referring video object segmentation methods under none, static and dynamic perturbations.

Method	Ref-YTVOS					
	Clean		Static		Dynamic	
	\mathcal{J}	\mathcal{F}	\mathcal{J}	\mathcal{F}	\mathcal{J}	\mathcal{F}
MTTR [1]	54.0	56.4	43.0 _{-9.0}	45.5 _{-10.9}	44.5 _{-9.5}	47.4 _{-9.0}
SgMg [8]	57.7	60.0	52.4 _{-5.3}	54.5 _{-5.5}	51.9 _{-5.8}	54.3 _{-5.7}
ReferFormer [10]	54.8	56.5	41.1 _{-13.7}	37.8 _{-18.7}	37.4 _{-17.8}	37.9 _{-18.6}
R ² -VOS [5]	56.1	58.4	45.3 _{-10.8}	47.4 _{-11.0}	44.9 _{-11.2}	46.8 _{-11.6}
OnlineRefer [9]	55.6	58.9	47.6 _{-8.0}	49.8 _{-9.1}	46.3 _{-9.3}	48.6 _{-10.3}

Table 14: Robustness of video object segmentation methods under low, medium, and high perturbations. The subscript $_s$ and $_u$ denote seen and unseen object categories respectively.

Method	YTVOS											
	Clean				Static				Dynamic			
	\mathcal{J}_s	\mathcal{F}_s	\mathcal{J}_u	\mathcal{F}_u	\mathcal{J}_s	\mathcal{F}_s	\mathcal{J}_u	\mathcal{F}_u	\mathcal{J}_u	\mathcal{F}_s	\mathcal{J}_u	\mathcal{F}_u
AOT [14]	83.9	88.8	79.9	88.5	79.0 _{-4.9}	83.1 _{-5.7}	72.4 _{-7.5}	80.5 _{-8.0}	77.2 _{-6.7}	81.0 _{-7.8}	68.9 _{-11.0}	76.7 _{-11.8}
DEAOT [16]	84.2	89.2	80.2	88.8	81.6 _{-2.6}	85.7 _{-3.5}	75.5 _{-4.7}	83.5 _{-5.0}	81.4 _{-2.8}	86.2 _{-3.0}	74.4 _{-5.8}	82.4 _{-6.4}
XMem [4]	84.3	88.6	80.3	88.6	79.5 _{-4.8}	83.4 _{-5.2}	74.4 _{-5.9}	82.3 _{-6.3}	77.5 _{-6.8}	81.2 _{-7.4}	70.9 _{-9.4}	78.0 _{-10.6}
DEVA [3]	85.0	89.4	79.7	88.0	81.8 _{-3.2}	86.1 _{-3.3}	75.7 _{-4.0}	83.7 _{-4.3}	79.8 _{-5.2}	83.8 _{-5.6}	71.6 _{-8.1}	79.2 _{-8.8}
Cutie [2]	85.6	90.0	80.6	88.3	81.6 _{-4.0}	85.7 _{-4.3}	75.5 _{-5.1}	83.5 _{-4.7}	79.7 _{-5.9}	83.7 _{-6.3}	71.6 _{-9.0}	79.0 _{-9.3}

Dataset	RefCOCO/+/g	DAVIS	YTVOS	Ref-DAVIS	Ref-YTVOS	AVS-s4	AVS-ms3
# Image	21928	3455	3859	3455	3455	10852	2120
# Text	126261	-	-	1200	12913	-	-
# Audio	-	-	-	-	-	10852	2120

Table 15: Statistics of Generated Datasets

4. **Speckle Noise.** Speckle noise is described through a multiplicative noise formula:

$$I' = I \cdot (1 + \rho \cdot \eta) \quad (3)$$

Here, ρ signifies the speckle noise’s intensity, and η is the Gaussian-distributed noise component.

Effects based on Blur. The act of blurring involves the convolution ($*$) of a source image I with a specific blur kernel K to produce a blurred output I' , represented as

$$I' = I * K \quad (4)$$

1. **Defocus Blur.** This mimics the blurring seen when an image is out of focus due to the camera lens’s characteristics, achieved by convolving the input image with a disc-shaped kernel.
2. **Glass Blur.** This simulates the look of viewing through a textured or patterned glass by utilizing an uneven kernel for convolution.
3. **Motion Blur.** Caused by swift movement of the camera or objects within the frame, this blur is modeled by convolving the image with a linear kernel aligned with the motion direction.

4. **Gaussian Blur.** This involves smoothing the image by convolving it with a Gaussian kernel, where the blur extent is governed by the kernel’s standard deviation.

Interference from the Environment. Environmental effects or weather conditions are often recreated using alpha blending methods. This technique merges a weather effect layer W with the base image I to formulate a mixed image I' as

$$I' = (1 - \alpha_W) \cdot I + \alpha_W \cdot W \quad (5)$$

with α_W dictating the mix ratio.

1. **Snow Effect.** The simulation of snow involves creating a layer filled with random white regions to mimic snowfall.
2. **Frost Effect.** By applying a semi-transparent whitening layer on the image, the frost effect is simulated, creating a weighted blend with the original image.
3. **Fog Effect.** This effect generates a foggy appearance by linearly interpolating between the original image and a uniform gray-scale image.
4. **Spatter Effect.** To replicate water droplets on a lens or window, a layer with semi-transparent dark marks is combined with the original image. A mask is applied to manage the effect’s presence in certain areas by designating transparent and affected regions.

Manipulations in Post-processing.

1. **Brightness.** This modification increases the image’s overall brightness by adding a brightness value b to each pixel (x, y) in I :

$$I'(x, y) = I(x, y) + b \quad (6)$$

2. **Contrast.** By scaling the image tones about the mean intensity \mathcal{J} , the contrast is adjusted:

$$I' = \alpha \cdot (I - \mathcal{J}) + \mathcal{J} \quad (7)$$

Here, α is the factor adjusting the contrast intensity.

3. **JPEG Compression.** This introduces the artifacts associated with the JPEG compression technique.
4. **Pixelate.** Reducing the image resolution, this effect aggregates pixels into blocks, assigning the average block value to each pixel within.

1.5 Implementation of Text Perturbations

Human-based issue. Human-based issues in text pertain to errors or variations commonly introduced by individuals during the writing or typing process.

1. **Misspelling.** For introducing misspellings throughout a text, we can conceptualize a function that probabilistically replaces correctly spelled words with their misspelled variants

$$T' = \begin{cases} T[i] & \text{if } p[i] > C \\ M(T[i]) & \text{otherwise} \end{cases}, i \in (1, m) \quad (8)$$

where m represents the number of words in text T , C is the probability threshold, and $M(\cdot)$ is the misspelling method that established based on LLM.

2. **Miss-punctuation.** To insert random punctuation into a text, we envision a function that introduces punctuation at probabilistic intervals within the text. The text transformation can be modeled as:

$$T' = \begin{cases} T[i] & \text{if } p[i] > C \\ P(T[i]) & \text{otherwise} \end{cases}, i \in (1, n) \quad (9)$$

where n represents the number of potential insertion points in text T , C is the probability threshold, and $P(\cdot)$ is the punctuation insertion method that operates based on a given severity level.

Sensor interference.

1. **Character Missing.** It occurs when characters are omitted from the text, potentially due to sensor errors, or transmission issues, which can be represented by

$$T' = \begin{cases} T[i] & \text{if } p[i] > C \\ \text{skip} & \text{otherwise} \end{cases}, i \in (1, n) \quad (10)$$

where C is the probability threshold and n is the length of character in text T .

1.6 Implementation of Acoustic Perturbations

Noise-based distortions. As we described in image-based perturbation, noise-based distortions in audio can be expressed as

$$A' = A + \eta \quad (11)$$

where A is the origin audio, η is the noise and when we sum them up, we can get an audio with noise A' .

1. **Gaussian Noise.** Gaussian noise is associated with a noise that is sampled from a Gaussian distribution with zero mean and variance σ^2 , written as $\eta \sim \mathcal{N}(0, \sigma^2)$.

2. **Background Noise.** Consider you are in a location with consistent ambient sounds, such as a cafe, an office, or outside with the sounds of nature or city life. Background noise can be considered any unwanted sound that interferes with the original audio signal. Unlike Gaussian noise, which is random and has a specific statistical distribution, background noise can be more complex and varied, encompassing a range of sounds from other human voices, machinery, wind, water, and more.
3. **Impulse Noise.** Impulse noise is characterized by sudden, short, and sharp sound spikes at irregular intervals. These noises are typically of a higher amplitude compared to the typical human-heard sound level.

Signal-level adjustment.

1. **Amplitude Gain.** Amplitude gain is the process of increasing the amplitude of an audio signal. It can be mathematically represented as

$$A' = G \cdot A \quad (12)$$

where G is the gain factor by which the signal's amplitude is multiplied.

2. **Tanh Distortion.** It applies the nonlinear tanh (hyperbolic tangent) function to the audio signal, creating a warm distortion effect, which can be written as

$$A' = \tanh A \quad (13)$$

Filtering effect. In audio processing, filtering effects involve selectively altering specific frequencies within an audio signal to achieve various outcomes, such as enhancing clarity, reducing noise, or shaping the tonal balance.

1. **Peak Filter.** Peak Filter boosts or cuts a specific frequency range around a center frequency. It can be represented by its effect on the frequency spectrum of the audio signal, altering the amplitude of the specified band.
2. **Low-pass Filter.** It Allows frequencies below a certain cutoff frequency to pass while attenuating frequencies above the cutoff. It's used to remove high-frequency noise or to simulate the effect of sounds being muffled.
3. **High-pass Filter.** Similar to the low-pass filter but allows frequencies above a certain cutoff frequency to pass while attenuating frequencies below the cutoff. It's used to remove low-frequency rumble or to make sounds appear thinner.

Physical interference.

1. **Air Absorption.** It Models the attenuation of sound as it travels through air, affecting frequencies differently based on distance. The equation can be summarized as

$$A'(f, d) = A(f) \cdot e^{-\alpha(f) \cdot d} \quad (14)$$

where $A'(f, d)$ is the amplitude of the frequency component f after traveling a distance d ; $A(f)$ is the original amplitude of the frequency component f ; $\alpha(f)$ is the absorption coefficient at frequency f ; and d is the distance the sound has traveled through the air.

2. **Room Reverberation.** It simulates the effect of sound reflecting off surfaces in an enclosed space, creating echoes and reverb, using the concept of convolution in signal processing. The equation for applying room reverberation through convolution is given by

$$A' = A \cdot I_{\text{room}} \quad (15)$$

where I_{room} is the impulse response of a specific room.

Post-processing manipulations.

1. **Time Mask.** Time Mask temporarily obscures a portion of the audio signal. The effect of a Time Mask can be represented by

$$A'[i] = \begin{cases} 0 & \text{if } i_{\text{start}} \leq i \leq i_{\text{end}} \\ A[i] & \text{otherwise} \end{cases} \quad (16)$$

where i_{start} and i_{end} are randomly selected indices within the audio signal’s duration by given proportion.

2. **MP3 Compression.** It Compresses the audio signal into the MP3 format, reducing file size with potential quality loss.

Table 16: Summary of considered tasks, metrics, referring modality, and datasets.

Task	RIS	VOS	R-VOS	AVS	Q3M
Inputs	Image, Text	Video, Mask	Video, Text	Video, Audio	Images, Text

1.7 Dataset Creation Details

In this section, we discuss the detailed data creation process. As shown in Tab. 16, we demonstrate the input types for each task. In our data creation process, we consider the noises that happened in all modalities including visual, textual and acoustic.

Dynamic and static perturbations. For image-level tasks, *i.e.*, RIS and Q3M, we just consider static perturbations where the perturbation type and severity do not change across time. For video-level tasks, we additionally consider a dynamic perturbation mode where the perturbation type and perturbation severity can change over time. To implement this, we treat each frame independently and randomly select perturbation type and severity with a uniform distribution.

Severity. For each perturbation type, we create 5 severity in the toolbox while only three of them are benchmarked in the R²-Bench. We attach the perturbation synthesis toolbox in the supplementary for detailed parameter settings for each perturbation type.

Perturbation type. In all dataset creation, we consider a maximum of 2 perturbations. The composition of perturbations follows the sequential order from source→environment→sensor→transmission. The noise types are selected randomly following a uniform distribution.

1.8 More Details about R²-Agent

In this section, we demonstrate the details of the user study for data selection evaluation and show more results of the model analysis.

Data selection evaluation. To evaluate the data selection performance of R²-Agent, we collect a dataset with paired perturbation types, evaluation metrics and data samples with corresponding human instruction. A total of 50 pairs are collected for the evaluation.

For the accuracy calculation, we separately consider each perturbation type, metrics and sample and calculate the accuracy between the human-annotated ones and R²-Agent selected ones.

For the human rating, we conduct a user study where the given instruction, all possible samples and R²-Agent selected samples are presented. The participants are asked to rate the rationality of the selection where 1 denotes reasonable and 0 means not. 7 participants participated in the user study. We average their rating across all samples as the Rate metrics.

More details of model analysis. To enable R²-Agent to analyze the performance, we first feed

- human instruction
- all perturbation types and corresponding explanations
- selected perturbations
- evaluation metrics
- captions of the evaluation set
- metrics of evaluation samples

to R²-Agent. After that, we give chain-of-thought prompts to encourage R²-Agent to analyze several hard-coded questions, including

- Performance change
- Case analysis
- Improvement recommendation.

Specifically, the output of the performance change analysis is fed as additional input to the improvement recommendation.

We demonstrate the prompts and corresponding responses in Fig. 2 and Fig. 3.

Prompt You are conducting a result analysis task. Given an instruction, perturbations and the evaluation metrics. You need to analyze the potential reason for the different performance changes. You may first analyze the perturbation types that have the largest and smallest impacts. For example, assuming the instruction is "Skiing on the mountain" and the results of J-index are {"motion blur": -12.5, "snow": -9.0}, the analysis can be "The motion blur perturbation has the largest impact on J-index suggesting the model fail to robustly segment the object in skiing scenario. The potential reason for the performance degradation can be due to 1) the motion blur can blur the object shape and appearance making the model failed to accurately perceive the object semantics and 2) the scenario of skiing are typically with snowing weather which introduces occlusions in the image. The further occlusion combined with motion blur can result in a severe reduction of imaging quality. For the snow perturbation, the model demonstrates a slightly mild performance degradation which can be because the skiing scenes have already had snow effect thus further snowing demonstrates a smaller performance drop."

Now you need to consider instruction: {driving scenario in a forest}, perturbations: {"frost": frost means the lens of the camera frozen, "fog": fog means the environment has fog weather, "spatter": spatter means a spatter occlusion on the camera.} and evaluation results: {"frost": -12.5, "fog": -9.0, "spatter": -7.2}. Can you analyze the results and provide an explanation?

Analysis: The evaluation results for the driving scenario in a forest with perturbations of frost, fog, and spatter are as follows:

- Frost: -12.5
- Fog: -9.0
- Spatter: -7.2

1.Frost: Frost perturbation demonstrates the largest impact on the evaluation metric. This suggests that the model's performance significantly deteriorates when the lens of the camera becomes frozen. Possible reasons for this degradation include:

1. Reduced Visibility: Frost on the lens can severely impair visibility, leading to blurred or distorted images. This impairment in visual clarity hampers the model's ability to accurately detect and identify objects in the driving scenario.
2. Altered Lighting Conditions: Frost may also affect the lighting conditions, causing glare or diffused light, which can confuse the model's perception of the scene.

2.Fog: Fog perturbation shows a substantial but slightly lesser impact compared to frost. This indicates that while foggy conditions negatively affect the model's performance, the impact is not as severe as frost. Possible explanations for this impact include:

1. Reduced Visibility: Similar to frost, fog also diminishes visibility by scattering light and obscuring objects in the environment. This reduction in visibility makes it challenging for the model to distinguish objects accurately.
2. Depth Perception Issues: Fog creates a sense of depth ambiguity, making it difficult for the model to accurately estimate distances between objects. This can lead to errors in object detection and localization.

3.Spatter: The spatter perturbation exhibits the smallest impact on the evaluation metric among the three perturbations. While still notable, the decrease in performance is not as significant as frost or fog. Potential reasons for this relatively mild impact include:

1. Partial Occlusion: Spatter occlusions on the camera introduce localized disruptions in the image rather than widespread impairment of visibility. As a result, the model may still retain some level of clarity in detecting objects despite the occlusions.
2. Adaptive Object Detection: The model may demonstrate a degree of robustness to spatter occlusions by adapting its detection strategies to account for partial obstructions. This adaptability allows the model to maintain a relatively stable performance compared to scenarios with more severe perturbations like frost or fog.

In summary, the driving scenario in a forest is significantly affected by perturbations such as frost and fog, which severely impair visibility and introduce challenges in object detection. Conversely, perturbations like spatter, while still impactful, result in a relatively smaller decrease in performance due to their localized nature and the model's potential adaptability to partial occlusions.

Fig. 2: Illustration of performance change analysis using R²-Agent.

Prompt: ...Now you need to consider instruction: (driving scenario in a forest), perturbations: {"frost": frost means the lens of the camera frozen, "fog": fog means the environment has fog weather, "spatter": spatter means a spatter occlusion on the camera.} and evaluation results: {"frost": -12.5, "fog": -9.0, "spatter": -7.2}. Can you analyze the results and provide an explanation?

Analysis: The evaluation results for the driving scenario in a forest with perturbations of frost, fog, and spatter are as follows...

Prompt: Based on the evaluation results and your analysis, can you give some improvement recommendation for each perturbation types?

Recomm.:

1.Frost:

- 1. Anti-frost Technology:** Incorporate anti-frost technology into the camera equipment used in the driving scenario. This technology can prevent the buildup of frost on the lens, ensuring clear and unobstructed vision even in cold and frost-prone environments.
- 2. Heating Mechanism:** Implement a heating mechanism for the camera lens to prevent frost formation. Heating elements can be strategically placed to maintain the lens temperature above freezing, thereby minimizing the risk of frost buildup during operation.
- 3. Dynamic Image Enhancement:** Develop algorithms for dynamic image enhancement that can effectively compensate for the distortion caused by frost. These algorithms can adjust image contrast, brightness, and sharpness in real-time to improve visibility and facilitate more accurate scene interpretation by computer vision models.

2.Fog:

- 1. Fog Penetration Algorithms:** Design algorithms specifically tailored to penetrate foggy conditions and enhance visibility in captured images. These algorithms can leverage advanced image processing techniques to mitigate the adverse effects of fog, such as dehazing and fog removal filters.
- 2. Multi-Sensor Fusion:** Integrate multiple sensors, such as LiDAR and radar, alongside visual cameras to complement image data with depth and environmental information. Fusion of data from different sensors can provide a more comprehensive understanding of the surroundings, enabling more robust perception and decision-making in foggy conditions.
- 3. Predictive Modeling:** Develop predictive models that anticipate the presence of fog based on environmental factors such as temperature, humidity, and terrain characteristics. By predicting foggy conditions in advance, adaptive strategies can be employed to mitigate their impact on camera visibility, such as adjusting camera settings or activating supplementary sensors.

3.Spatter:

- 1. Automated Cleaning Mechanism:** Implement an automated cleaning mechanism for the camera lens to remove spatter occlusions during operation. This mechanism can utilize techniques such as air blasts or wiper systems to periodically clear the lens surface and maintain optimal visibility.
- 2. Hydrophobic Coatings:** Apply hydrophobic coatings to the camera lens to repel water and prevent spatter accumulation. These coatings create a protective barrier that reduces adhesion of water droplets and other contaminants, minimizing the need for manual cleaning and improving long-term performance in wet conditions.
- 3. Real-Time Occlusion Detection:** Develop algorithms capable of detecting and identifying spatter occlusions in real-time. By actively monitoring the cleanliness of the camera lens, these algorithms can trigger cleaning actions or adjust processing strategies to mitigate the impact of spatter on image quality and model performance.

By implementing these improvement recommendations tailored to each perturbation type, the robustness and performance of computer vision systems in driving scenarios within forest environments can be significantly enhanced, enabling more reliable and accurate interpretation of the surrounding environment for various applications such as autonomous driving and advanced driver assistance systems

Fig. 3: Illustration of solution recommendation analysis using R²-Agent.

References

1. Botach, A., Zheltonozhskii, E., Baskin, C.: End-to-end referring video object segmentation with multimodal transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4985–4995 (2022) [6](#)
2. Cheng, H.K., Oh, S.W., Price, B., Lee, J.Y., Schwing, A.: Putting the object back into video object segmentation. arXiv preprint arXiv:2310.12982 (2023) [2](#), [6](#)
3. Cheng, H.K., Oh, S.W., Price, B., Schwing, A., Lee, J.Y.: Tracking anything with decoupled video segmentation. In: ICCV (2023) [3](#), [6](#)
4. Cheng, H.K., Schwing, A.G.: Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In: European Conference on Computer Vision. pp. 640–658. Springer (2022) [6](#)
5. Li, X., Wang, J., Xu, X., Li, X., Raj, B., Lu, Y.: Robust referring video object segmentation with cyclic structural consensus. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22236–22245 (2023) [5](#), [6](#)
6. Liu, J., Ding, H., Cai, Z., Zhang, Y., Satzoda, R.K., Mahadevan, V., Manmatha, R.: Polyformer: Referring image segmentation as sequential polygon generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18653–18663 (2023) [2](#), [3](#), [4](#)
7. Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A.L., Murphy, K.: Generation and comprehension of unambiguous object descriptions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 11–20 (2016) [2](#)
8. Miao, B., Bennamoun, M., Gao, Y., Mian, A.: Spectrum-guided multi-granularity referring video object segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 920–930 (2023) [4](#), [5](#), [6](#)
9. Wu, D., Wang, T., Zhang, Y., Zhang, X., Shen, J.: Onlinerefer: A simple online baseline for referring video object segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2761–2770 (2023) [4](#), [5](#), [6](#)
10. Wu, J., Jiang, Y., Sun, P., Yuan, Z., Luo, P.: Language as queries for referring video object segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4974–4984 (2022) [5](#), [6](#)
11. Xu, N., Yang, L., Fan, Y., Yue, D., Liang, Y., Yang, J., Huang, T.: Youtube-vos: A large-scale video object segmentation benchmark. arXiv preprint arXiv:1809.03327 (2018) [2](#)
12. Xu, Z., Chen, Z., Zhang, Y., Song, Y., Wan, X., Li, G.: Bridging vision and language encoders: Parameter-efficient tuning for referring image segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17503–17512 (2023) [2](#), [3](#), [4](#)
13. Yang, Z., Wang, J., Tang, Y., Chen, K., Zhao, H., Torr, P.H.: Lavt: Language-aware vision transformer for referring image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18155–18165 (2022) [2](#), [3](#), [4](#)
14. Yang, Z., Wei, Y., Yang, Y.: Associating objects with transformers for video object segmentation. Advances in Neural Information Processing Systems **34**, 2491–2502 (2021) [6](#)
15. Yang, Z., Yang, Y.: Decoupling features in hierarchical propagation for video object segmentation. Advances in Neural Information Processing Systems **35**, 36324–36336 (2022) [4](#)
16. Yang, Z., Yang, Y.: Decoupling features in hierarchical propagation for video object segmentation. Advances in Neural Information Processing Systems **35**, 36324–36336 (2022) [6](#)

17. Yu, L., Poirson, P., Yang, S., Berg, A.C., Berg, T.L.: Modeling context in referring expressions. In: European Conference on Computer Vision. pp. 69–85. Springer (2016) [2](#)
18. Zou, X., Dou, Z.Y., Yang, J., Gan, Z., Li, L., Li, C., Dai, X., Behl, H., Wang, J., Yuan, L., et al.: Generalized decoding for pixel, image, and language. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15116–15127 (2023) [2](#), [3](#), [4](#)
19. Zou, X., Yang, J., Zhang, H., Li, F., Li, L., Wang, J., Wang, L., Gao, J., Lee, Y.J.: Segment everything everywhere all at once. *Advances in Neural Information Processing Systems* **36** (2024) [2](#), [3](#), [4](#)