Supplementary Material: Revisiting Visual Corruptions in LVLMs: A Shape–Texture Perspective on Model Failures

Xinkuan Qiu[1,3,5], Meina Kan[2], Zhenliang He [2], Yongbin Zhou[1,4], Shiguang Shan[2]

[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, 100085, China

[2]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China

[3]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, 100049, China

[4]School of Cyber Science and Engineering, Nanjing University of Science and Technology, Nanjing, 210094, China

[5]State Key Laboratory of Cyberspace Security Defense, Beijing, 100085, China

qiuxinkuan@iie.ac.cn, {kanmeina, hezhenliang, sgshan}@ict.ac.cn, zhouyongbin@njust.edu.cn

A. Dataset

A1. Design of RWIC-VQA Dataset

Synthetic corruptions are widely used in robustness benchmarks such as ImageNet-C [4] and COCO-C [8], as well as their extensions to video and multimodal settings [13, 17]. While these distortions offer controlled evaluation settings, they do not reflect the complex degradation patterns found in practical, real-world scenarios. As a result, current robustness evaluations of LVLMs greatly underrepresent challenges introduced by naturally occurring corruptions.

To address this gap, we introduce **RWIC-VQA**(Real-World Image Corruption VQA), a benchmark specifically designed to evaluate LVLM robustness under real-world corruptions. RWIC-VQA contains **1,153 manually verified VQA samples**, spanning five common categories of real degradation: *noise, blur, rain, snow, and haze*. Representative samples are shown in Figure 1. The construction pipeline consists of three steps:

Step 1: Image Collection. We collect real-world corrupted images from publicly available restoration datasets, including: MCWNNM [15], SIDD [1], RealBlur [14], RWBI [18], RainDrop [12], REVIDE [19], and RSOD [3]. These datasets cover diverse noise processes, optical degradation, water droplets, adverse weather conditions, and atmospheric scattering.

Step 2: QA Generation. For each image in the collected datasets, we employ GPT-4o [11] to generate multiple-choice QA pairs following the MMBench [9] format. When clean paired images exist, they are also provided to improve question relevance and correctness. The exact prompt used for generation is preserved below:

> *We need to design a suitable visual question for this image. Please refer to the following perspectives. If any are applicable to the image, generate a question and provide multiple-choice answers including incorrect ones. The format reference is as follows:*
>
> 1. *Image Classification: Determine the main subject of the image.*
> 2. *Object Localization: For a single object, determine its position in the image (such as top, bottom, etc.), its absolute/relative location.*
> 3. *Attribute Recognition: Recognition of texture, shape, appearance characteristics, emotions.*
> 4. *OCR: Recognition of text, formula, and sheet in the image.*
> 5. *Spatial Relationship: Determine the relative position between objects in the image.*
> 6. *Attribute Comparison: Compare attributes of different objects in the image, such as shape, color, etc.*
> 7. *Action Recognition: Recognizing human actions, including pose motion, human-object interaction, and human-human interaction.*
>
> *Format reference for generation — it must be a structure that can be parsed by* `json.loads`.

Step 3: Human Verification. All QA pairs undergo manual verification. Counting-related and relational questions, which often exhibit generation errors, are carefully corrected. This ensures the final benchmark is reliable even under severe corruptions.

By integrating naturally corrupted images with diverse question types, RWIC-VQA provides a realistic robustness evaluation protocol complementary to synthetic corruption benchmarks.

A2. Illustrative QA Pairs from Datasets Evaluated

To demonstrate the diversity and difficulty of the benchmarks used in our evaluations, Figures 1–4 show representative QA samples from RWIC-VQA, ImageNet10-C, MMBench-C, and POPE-MSCOCO-C. These examples cover tasks including classification, spatial reasoning, OCR, attribute understanding, and hallucination detection, reflecting the broad skill set required for robust multimodal understanding under corruption.

B. Experiments

B1. Computing Environment

All experiments were conducted on a workstation equipped with four NVIDIA RTX 3090 GPUs, using CUDA 12.2 and Python 3.9.21. For LLaVA-1.5 [7], mPLUG-Owl2 [16], and Qwen-VL [2], we followed the official repositories and integrated additional dependencies from the VCD framework [6]. This unified setup ensures reproducibility and consistent evaluation across all models.

B2. Method Implementation

For methods with publicly available implementations (e.g., VCD), we used the original code from the corresponding GitHub repositories. For methods lacking public code, we reproduced the implementations within the VCD framework to ensure consistent interfaces and fair comparison. We cache visual features once (orig/shape/texture); decoding only adds extra language-decoder forward passes to get/fuse logits (no per-step image re-encoding). All hyperparameters were set according to the default configurations reported in the respective papers, ensuring optimal and comparable performance across methods. All experiments were conducted once, as the use of a fixed random seed (`seed=42`) eliminates variability and ensures deterministic behavior. To assess robustness to stochasticity, we rerun ST-CD with LLaVA on ImageNet10-C for 5 runs and obtain 84.04±0.21 average accuracy.

B3. Inference efficiency and latency

We report absolute inference time on ImageNet10-C (20k images) to assess deployment feasibility (Table 1). All methods are evaluated under the same setting on 2× RTX 3090. ST-CD incurs a moderate overhead over VCD while remaining substantially cheaper than VACoDe. The additional cost mainly comes from computing one extra contrastive view: VCD uses two views(original + noised), whereas ST-CD uses three (original + edge + jigsaw), leading to an empirical latency of ∼1.5× VCD, consistent with the view count. In this setting, ST-CD takes 0.564 s/image, which is practical for interactive settings. As a reference, a classic rule-of-thumb suggests that response times within ∼1s typically keep users' flow of thought uninterrupted, whereas ∼10s risks losing attention and requires feedback [10].

Method	LLaVA-1.5	mPLUG-Owl2	Qwen-VL	Avg	x VCD
VCD	0.377	0.360	0.372	0.370	1.0
VACoDe	1.404	1.578	1.680	1.554	4.2
Ours	0.570	0.522	0.600	0.564	1.5

Table 1. Inference time (second per image).

B4. Experiments in broader settings.

(i) **Mixed corruptions.** ST-CD naturally extends to mixed corruptions. For a single corruption (e.g., noise), the texture probe typically provides a more reliable contrastive signal and is upweighted by the entropy-based fusion, while the shape branch is downweighted when its logits are less informative. For mixed corruptions (e.g., noise+blur), the degradation can be viewed as simultaneous perturbations along both the shape and texture axes in our perceptual subspace; ST-CD therefore combines the two corrections additively to compensate both biases (e.g., reducing shifts toward texture-similar or shape-similar confusions such as *bear* or *wolf* for a *dog* image). Empirically, on ImageNet10-C with LLaVA, ST-CD yields consistent gains under mixed corruptions (Table 2).

(ii) **Additional corruption families.** Beyond the four primary families (noise/blur/geometric/color), RWIC-VQA includes real-world weather effects (rain/haze/snow), where ST-CD consistently improves robustness. We further evaluate occlusion (Cutout/CoarseDropout) and compression (JPEG/pooling), and ST-CD continues to provide gains (Table 3).

(iii) **Longer-text generation.** Since ST-CD is a decoding-time calibration method, it can be directly applied to open-ended generation; we address the long-form generation concern by reporting LLaVA's CHAIR results on **COCO Captions** with synthetic corruptions (Table 3).

Method	N+B	B+N	C+G	G+C	N+G	G+N	C+B	B+C	Avg
Baseline	49.5	25.1	51.7	55.7	42.9	41.7	44.8	52.2	45.5
ICD	59.3	30.0	59.4	66.3	48.8	52.3	53.7	60.4	53.8
VCD	60.6	28.3	58.4	66.2	51.4	51.6	**55.5**	62.2	54.3
LCD	60.6	33.7	60.9	67.5	50.9	53.2	54.8	61.0	55.3
VACoDe	61.6	34.0	59.5	67.1	**54.1**	54.9	52.9	59.8	55.5
Ours	**61.9**	**34.5**	**63.6**	**70.2**	53.9	**55.3**	54.6	**64.4**	**57.3**

Table 2. Results on mixed corruptions.

B5. Hyperparameter Settings for Learned Weighting Strategy

For the learned weighting strategy (Sec. 6.2 of the main paper), we train a lightweight predictor to estimate fusion coefficients (α_s, α_t) based on frequency-domain represen-

Method	(ii) Additional families		(iii) Longer text generation	
	Occlusion	Compression	$CHAIR_S \downarrow$	$CHAIR_I \downarrow$
Baseline	81.7	62.7	27.95	9.36
ICD	92.2	73.0	24.86	7.92
VCD	92.9	73.5	26.18	8.14
LCD	94.7	75.2	23.42	7.81
VACoDe	94.7	74.9	24.09	7.70
Ours	**95.3**	**75.5**	**22.10**	**6.89**

Table 3. Results on broader corruption and longer text generation.

tations. This module is used only for ablation; the main results use the training-free entropy weighting scheme.

- **Training data:** 30% of ImageNet10-C, stratified across corruption type and severity.
- **Input processing:** resize to 224×224, grayscale, 2D FFT magnitude spectrum.
- **Model:** ResNet-18 with 1-channel input; final FC layer outputs a 2D softmax vector.
- **Optimizer:** Adam, lr = 1×10^{-3}, $(\beta_1, \beta_2) = (0.9, 0.999)$; LR halved at epochs 100 and 150.
- **Batch size:** 64
- **Epochs:** 200

C. Codes

C1. Corruption Generation Code Details

To ensure full reproducibility of our corruption-based evaluations, we provide detailed implementation procedures for generating the corrupted images used throughout the experiments. We adopt eight widely used synthetic corruption types grouped into four categories, following the standard taxonomy defined by ImageNet-C:

- **Noise**: Gaussian Noise, Speckle Noise
- **Blur**: Defocus Blur, Motion Blur
- **Color Distortion**: Channel Shuffle, Inversion
- **Geometric Distortion**: Shearing, Elastic Transformation

All corruptions are implemented using the `imgaug` Python library [5]. We utilize predefined augmentation classes with controlled severity parameters to ensure consistency and comparability across corruption types. The following code snippet demonstrates how to apply a specific corruption to a single image:

```
corrupted_image = iaa.imgcorruptlike.
    ElasticTransform(severity = 5)(image=
    image)
corrupted_image = iaa.imgcorruptlike.
    GaussianNoise(severity = 5)(image=image)
corrupted_image = iaa.imgcorruptlike.
    SpeckleNoise(severity = 5)(image=image)
corrupted_image = iaa.imgcorruptlike.
    MotionBlur(severity = 5)(image=image)
corrupted_image = iaa.imgcorruptlike.
    DefocusBlur(severity = 5)(image=image)
corrupted_image = iaa.ShearY(45*random.choice
    ([1, -1]))(image=image)
corrupted_image = iaa.ChannelShuffle(1.0)(
    image=image)
corrupted_image = iaa.Invert(1.0)(image=image)
```

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. A high-quality denoising dataset for smartphone cameras. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1692–1700, 2018. 1

[2] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, and et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 2

[3] Qiqi Ding, Peng Li, Xuefeng Yan, Ding Shi, Luming Liang, Weiming Wang, Haoran Xie, Jonathan Li, and Mingqiang Wei. Cf-yolo: Cross fusion yolo for object detection in adverse weather with a high-quality real snow dataset. *IEEE Transactions on Intelligent Transportation Systems*, 24(10): 10749–10759, 2023. 1

[4] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *International Conference on Learning Representations*, 2018. 1

[5] Alexander B. Jung, Kentaro Wada, Jon Crall, et al. imgaug: Image augmentation library. https://github.com/aleju/imgaug, 2020. 3

[6] Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13872–13882, 2024. 2

[7] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 2024. 2

[8] Jiawei Liu, Zhijie Wang, Lei Ma, Chunrong Fang, Tongtong Bai, Xufan Zhang, Jia Liu, and Zhenyu Chen. Benchmarking object detection robustness against real-world corruptions. *International Journal of Computer Vision*, pages 1–19, 2024. 1

[9] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, and Yike Yuan et al. Mmbench: Is your multi-modal model an all-around player? *European Conference on Computer Vision*, pages 216–233, 2025. 1

[10] Jakob Nielsen. *Usability Engineering*. Morgan Kaufmann, San Francisco, CA, 1993. 2

[11] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, and Lama Ahmad et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2024. 1

[12] Rui Qian, Robby T. Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2482–2491, 2018. 1

[13] Jielin Qiu, Yi Zhu, Xingjian Shi, Florian Wenzel, Zhiqiang Tang, Ding Zhao, Bo Li, and Mu Li. Benchmarking robustness of multimodal image-text models under distribution shift. *Journal of Data-centric Machine Learning Research*, 2023. 1

[14] Jaesung Rim, Haeyun Lee, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking de-

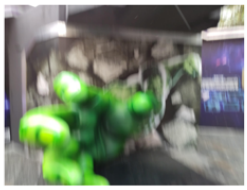
Corruption type: Noise

Question: *What is the main purpose of the chart depicted in the image?*
Choices:
A. To measure light intensity
B. To classify different shades of gray
C. To display a color palette for selection
D. To show a range of monochromatic shades

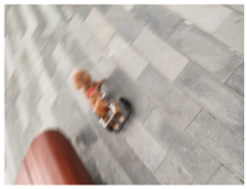
Question: *What is the title of the book by authors Russell and Norvig visible in the image?*
Choices:
A. Design Patterns
B. 3D Graphics Programming
C. Artificial Intelligence: A Modern Approach
D. Introduction to Java Programming

Question: *What is the main color of the smallest toy truck in the image?*
Choices:
A. Red
B. Yellow
C. Blue
D. Green

Corruption type: Blur



Question: *What is the character of the large object in the foreground?*
Choices:
A. Superman
B. Captain America
C. Hulk
D. Batman

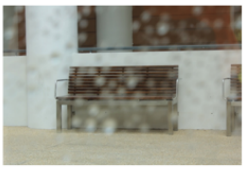


Question: *What is the main action occurring in the image?*
Choices:
A. A dog is fetching a ball.
B. A dog in a wheelchair is moving.
C. A dog is sitting.
D. A dog is running.

Question: *How many chairs are visible around one table in the image?*
Choices:
A. Two
B. Five
C. Three
D. Four

Corruption type: Rain

Question: What is the main subject of the image?
Choices:
A. A residential house
B. A public park
C. A commercial building
D. A school



Question: What is the primary color of the banners attached to the poles closest to the camera?
Choices:
A. Yellow
B. Blue
C. Green
D. Red

Question: Where is the car located in the parking area?
Choices:
A. Near the entrance on the right side
B. On the left side near the building
C. At the back of the parking area
D. In the middle of the parking lot

Corruption type: Haze

Question: What object is located on top of the cupboard in the image?
Choices:
A. A lamp
B. A cardboard box
C. A potted plant
D. A vase

Question: What object is placed on the table in the lower left corner of the image?
Choices:
A. A small lamp
B. A stack of books
C. A set of color cards
D. A vase of flowers

Question: What is the primary function of the device mounted on the wall in the center of the image?
Choices:
A. Air conditioner
B. Oven
C. Television
D. Water heater

Corruption type: Snow

Question: *What is the dominant color of the car closest to the camera?*
Choices:
A. Green
B. Blue
C. Black
D. Red

Question: *How many cars are visible on the snowy road?*
Choices:
A. Four
B. Five
C. Two
D. Three

Figure 1. Representative VQA examples from RWIC-VQA across five corruption types

Defocus Blur

Shearing

Speckle Noise

Invert

Zoom Blur

Elastic Transformation

Gaussian Noise

Channel Shuffle

Question: *What is the main object in the image above?*
Choices: [green lizard, goldfinch, ring-tailed lemur, llama, samoyed, race car, ocean liner, corkscrew, cd player, snake fence].
Answer (must be exactly one of the choices):

Figure 2. Representative samples from the ImageNet10-C dataset, illustrating diverse synthetic corruption types and class prompts.

Shear

Hint: Figure: Great Victoria Desert.
The Great Victoria Desert is a hot desert ecosystem located in Western Australia and South Australia. It is the largest desert in Australia! The Great Victoria Desert is home to the rare great desert skink. To stay cool during the day, great desert skinks live in holes they dig in the ground.

Question: *Which statement describes the Great Victoria Desert ecosystem?*

Options:
A. It has thick, moist soil.
B. It has dry, thin soil.

Gaussian Noise

Hint: The passage below describes an experiment. Read the passage and then follow the instructions below.
Madelyn applied a thin layer of wax to the underside of her snowboard and rode the board straight down a hill. Then, she removed the wax and rode the snowboard straight down the hill again. She repeated the rides four more times, alternating whether she rode with a thin layer of wax on the board or not. Her friend Tucker timed each ride. Madelyn and Tucker calculated the average time it took to slide straight down the hill on the snowboard with wax compared to the average time on the snowboard without wax.
Figure: snowboarding down a hill.
Question: *Identify the question that Madelyn and Tucker's experiment can best answer.*
Options:
A. Does Madelyn's snowboard slide down a hill in less time when it has a thin layer of wax or a thick layer of wax?
B. Does Madelyn's snowboard slide down a hill in less time when it has a layer of wax or when it does not have a layer of wax?

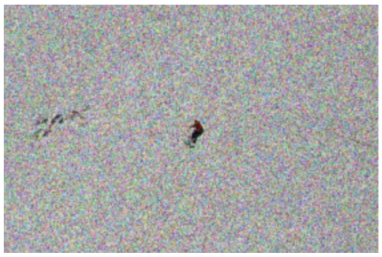
Figure 3. Representative QA examples from MMBench-C, showing task diversity and multimodal reasoning capabilities under corruption.

Defocus Blur

Question_1: *Is there a truck in the image?*
Question_2: *Is there a pizza in the image?*
Question_3: *Is there a person in the image?*
Question_4: *Is there a sink in the image?*
Question_5: *Is there an umbrella in the image?*
Question_6: *Is there a cell phone in the image?*

Invert

Question_1: *Is there a person in the image?*
Question_2: *Is there a sheep in the image?*
Question_3: *Is there a bicycle in the image?*
Question_4: *Is there a broccoli in the image?*
Question_5: *Is there a car in the image?*
Question_6: *Is there a tennis racket in the image?*



Speckle Noise

Question_1: *Is there a snowboard in the image?*
Question_2: *Is there a car in the image?*
Question_3: *Is there a person in the image?*
Question_4: *Is there a sandwich in the image?*
Question_5: *Is there a skis in the image?*
Question_6: *Is there a couch in the image?*

Figure 4. Representative QA examples from POPE-MSCOCO-C, focused on object existence and hallucination under corrupted conditions.

blurring algorithms. *European Conference on Computer Vision*, pages 184–201, 2020. 1

[15] Jun Xu, Lei Zhang, David Zhang, and Xiangchu Feng. Multi-channel weighted nuclear norm minimization for real color image denoising. *IEEE International Conference on Computer Vision*, pages 1096–1104, 2017. 1

[16] Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, and Fei Huang. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13040–13051, 2024. 2

[17] Jiawei Zhang, Tianyu Pang, Chao Du, Yi Ren, Bo Li, and Min Lin. Benchmarking large multimodal models against common corruptions. *arXiv preprint arXiv:2401.11943*, 2024. 1

[18] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Bjorn Stenger, Wei Liu, and Hongdong Li. Deblurring by realistic blurring. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2737–2746, 2020. 1

[19] Xinyi Zhang, Hang Dong, Jinshan Pan, Chao Zhu, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Fei Wang. Learning to restore hazy video: A new real-world dataset and a new method. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9239–9248, 2021. 1