Enhancing Vision Language Corruption Robustness using Cross-Distribution & Prompted Denoisers

Sameer Shafayet Latif[1*], Sadab Shiper[1*], K. M. Rahiduzzaman Kiran[1*], Md Farhan Ishmam[2*]
Md Azam Hossain[1], Abu Raihan Mostofa Kamal[1], Md Hamjajul Ashmafee[3]

[1]*Islamic University of Technology* [2]*University of Utah* [3]*University of Alberta*

[*]Equal Contribution {sameershafayet,farhanishmam,ashmafee}@iut-dhaka.edu

Abstract

While the current generation of Vision Language Models (VLMs) has excelled in ideal conditions, their performance drops significantly when exposed to realistic multimodal corruptions, such as blurry images and grammatically incorrect text. Our work addresses this by establishing a novel multimodal corruption and denoising benchmark, VLSRB, with a rich suite of 18 visual and 18 textual corruption functions, to evaluate the system robustness of VLMs. To enhance robustness, we employ: (i) cross-distribution visual denoisers, inspired by the Mixture of Experts (MoE) architecture, and (ii) a prompted zero-shot textual denoiser. Our experiments reveal an overall accuracy gain of up to 5.5%, while revealing the vulnerability of models to specific corruptions and their over-reliance on the textual modality. We envision that the behavioral insights from our benchmark will help in developing robust VLM systems. Our code is available at: https://github.com/farhanishmam/VLMDenoising.

1. Introduction

Vision Language Models (VLMs) [27, 50, 91] have achieved strong performance in several downstream tasks, *e.g.*, Visual Question Answering [2, 39], Image Captioning [79, 85], Image-Text Retrieval [66], and Visual Grounding [87]. These models underpin a wide range of real-world systems, *e.g.*, medical systems [53, 73], autonomous driving systems [16], and visual chatbots [20]. With increasing use cases in the real world, VLMs are more susceptible to corruptions in visual [40], textual [37], or both modalities [92], which can significantly degrade the overall performance.

The set of existing multimodal corruption robustness benchmarks [17, 54, 92] is concerned with model robustness, *i.e.*, the model's inherent ability to maintain reliable performance under corrupted inputs. We draw a sharp contrast and define system robustness [42, 72] in the context of multimodal corruption effects.

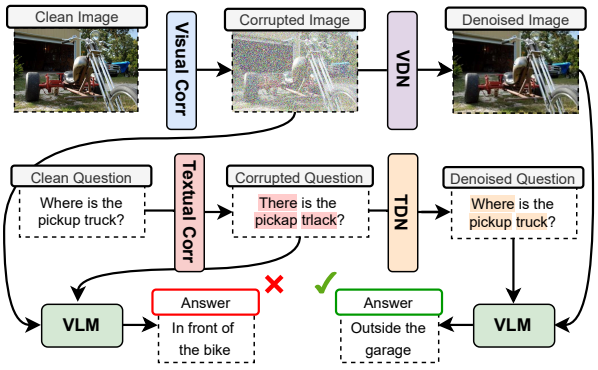


Figure 1. Overview of our denoising framework. VLMs fail to answer correctly when exposed to common visual and textual corruptions. The denoised image & question highlights the contribution of denoisers to robustness enhancement of VLMs by effectively suppressing noise-induced distortions, ensuring reliable features for improved VQA performance.

> ***System robustness*** *is the degree to which an end-to-end system's (including a model, pre- and post-processing units) task performance is preserved when one or more input modalities are corrupted.*

System robustness can be deemed more critical than model robustness during deployment, as end-to-end performance matters more than the model's inference capabilities [4]. Hence, benchmarks that only evaluate model robustness can be inadequate for practical usability.

We address this by introducing a benchmark to evaluate the system robustness of VLMs and enhance it using plug-and-play denoisers. Our visual denoising framework draws inspiration from the Mixture of Experts [25] architecture to ensure generalization in a cross-distribution setting. The merit of the textual denoiser

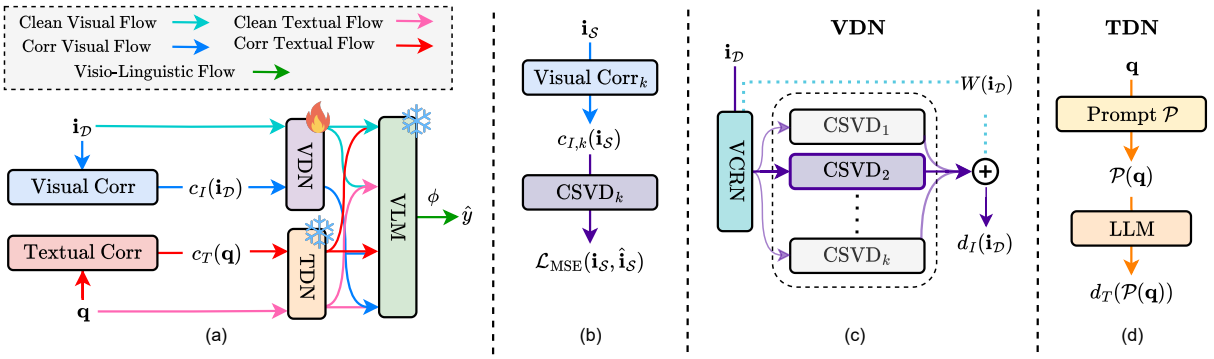


Figure 2. (a) The baseline methods used in our VLM benchmark on target distribution image $\mathbf{i}_\mathcal{D}$ and questions $\mathbf{q}$, using the visual and textual corruptions c_I and c_T respectively. ϕ maps the VLM generated answer to the answer class $\hat{y}$. (b) The training paradigm of the Corruption Specific Visual Denoiser (CSVD) module of the Visual DeNoiser (VDN) where images from a source distribution $\mathbf{i}_\mathcal{S}$ in a specific visual corruption $c_{I,k}$ is used to train a specific CSVD$_k$, (c) Inference of the VDN d_I on target distribution image $\mathbf{i}_\mathcal{D}$ in two stages using the Visual Corruption Routing Network (VCRN) and a specific CSVD to produce the denoised image, (d) Inference of the Textual DeNoiser (TDN) d_T using an LLM. Only the VDN is trainable (fire symbol), while the TDN and VLM are frozen (snowflake symbol).

lies in its simplicity, particularly in zero-shot prompting. We summarize our contributions as follows:

1. We establish a novel Vision-Language System Robustness Benchmark (VLSRB) with a suite of 36 multimodal corruption effects (18 visual & 18 textual), mimicking real-world noisy conditions.
2. We introduce cross-distribution visual denoisers and prompted zero-shot textual denoisers to enhance multimodal robustness and improve accuracy in VQA up to 5.5% overall and 9% in certain categories.
3. Our findings reveal several behavioral insights, *e.g.*, over-reliance on the textual modality and the vulnerability to certain corruption categories.

2. Related Work

Robustness of VLMs can have several categories: *adversarial robustness* entails defense against worst-case, tailored, and/or imperceptible perturbations to disrupt the model prediction [6], *corruption robustness* quantifies the model's resilience to realistic noise, occlusions, or distortions [60], and *out-of-distribution robustness* quantifies the model's generalization capabilities to new data distributions [34]. System robustness has been explored in the broader contexts as the measure of persistence against perturbations in a system [42, 72]. Our work addresses the system robustness of VLMs in the context of corruption robustness, which has been explored in visual [40, 78], textual [10, 36], and multimodal [65, 92] settings.

VL Robustness Enhancement include adversarial training [9, 100], randomized smoothing [15, 90], adaptation methods [12], contrastive learning [86], and data augmentation [67]. In contrast, our work focuses on enhancing robustness via denoising, which has achieved remarkable performance with the current generation of VLMs [38, 43].

Visual and Textual Denoising. Image denoising has been central to vision tasks, *e.g.*, image restoration [11, 18, 52], low-light image enhancement [8], and medical imaging [28, 62]. Deep learning approaches, particularly CNN-based [88, 93, 94] and transformer-based architectures [55, 89] have outperformed classical methods [5]. For visual denoising, our work relies on off-the-shelf CNN-based denoisers, such as DRUNet [41], but in a multi-corruption setting. Textual denoising has been used for OCR [69], contextual text denoising [75], machine translation [58], and text generation [82]. For textual denoising, we prompt LLMs as denoisers [98].

3. Background

3.1. Classification vs Generative VQA

The VQA task has been traditionally defined as a classification problem where a classifier $f_\theta(\cdot)$ maps the joint image-question feature space to an answer class, *i.e.*, $f_\theta : \mathcal{I} \times \mathcal{Q} \rightarrow \mathcal{Y}$. For the image representation $\mathbf{i} \in \mathbb{R}^{d_I}$ and question representation $\mathbf{q} \in \mathbb{R}^{d_Q}$, with d_I and d_Q representing the associated vector dimensions, the predicted answer class can be formulated as,

$$\hat{y} = f_\theta(\mathbf{i}, \mathbf{q}). \tag{1}$$

We instead define VQA as the answer generation task on the same visual representation $\mathbf{i}$ and textual question with additional prompting tokens $\mathbf{t}$, for predicting an an-

swer sequence, $\mathbf{a} = \langle a_1, a_2, ...a_l \rangle$, such that,

$$a_i = \arg\max \mathbb{P}(a|\mathbf{i}, \mathbf{t}, a_{<i}). \quad (2)$$

We denote the generative VQA model as $g_\theta(\cdot)$ and define an extraction function $\phi(\cdot)$ that maps the free-form generated answer $\mathbf{a}$ to an answer class y, s.t, $\phi : \mathcal{A} \rightarrow \mathcal{Y}$. The associated answer class prediction is,

$$\hat{y} = \phi(\mathbf{a}) = \phi(g_\theta(\mathbf{i}, \mathbf{t})). \quad (3)$$

3.2. Corruption Robustness

Following previous works [33, 40], we define the expected robustness in both visual and textual modalities. We denote visual and textual corruption functions as $c_I \in \mathcal{C}_I$ and $c_T \in \mathcal{C}_T$ respectively, with the associated real life occurrence probabilities, $\mathbb{P}_{\mathcal{C}_I}(c_I)$ and $\mathbb{P}_{\mathcal{C}_T}(c_T)$. The test accuracy on target distribution $\mathcal{D}$ of the generative VQA model g_θ can be derived as $\mathbb{P}_{(\mathbf{i},\mathbf{t},y)\sim\mathcal{D}}(\phi(g_\theta(\mathbf{i}, \mathbf{t})) = y)$, following Eq. (3). Hence, the corruption robustness can be formulated as the average case performance,

$$\mathbb{E}_{c_I\sim\mathcal{C}_I, c_T\sim\mathcal{C}_T}[\mathbb{P}_{(\mathbf{i},\mathbf{t},y)\sim\mathcal{D}}(\phi(g_\theta(c_I(\mathbf{i}), c_T(\mathbf{t}))) = y)]. \quad (4)$$

4. Methodology

Following Fig. 2, we enhance vision-language robustness using two separate networks in each modality: (i) Cross-Distribution Visual Denoiser and (ii) Zero-shot Textual Denoiser.

4.1. Cross-Distribution Visual Denoising

Given, the source image distribution $\mathcal{I}_S$ and target image distribution $\mathcal{I}_\mathcal{D}$, we define the visual denoiser,

$$d_I : \bigcup_{c_I\in\mathcal{C}_I} c_I(\mathcal{I}_S) \rightarrow \mathcal{I}_S. \quad (5)$$

Ideally, visual denoising should be a close approximation of the inverse of each of the visual corruption functions, *i.e.*,

$$d_I(c_I(\mathbf{i})) \approx c_I^{-1}(c_I(\mathbf{i})) = \mathbf{i} \quad \forall c_I \in \mathcal{C}_I, \mathbf{i} \sim \mathcal{I}_S. \quad (6)$$

We formulate the cross-distribution test performance as,

$$\mathbb{E}_{\mathbf{i}\sim\mathcal{I}_\mathcal{D}, c_I\sim\mathcal{C}_I}[||d_I(c_I(\mathbf{i})) - \mathbf{i}||^2]. \quad (7)$$

4.2. Visual DeNoiser (VDN) Module

The VDN architecture is conceptually similar to sparse expert networks [25]. The VDN is composed of two modules: (i) Visual Corruption Routing Network (VCRN) and (ii) Corruption-Specific Visual Denoiser (CSVD). VCRN and CSVD function analogously to the routers and experts in a sparse expert model. The VCRN directs the image to a set of CSVD experts, and each CSVD reconstructs the image for a specific corruption class. The final image is the weighted sum of the individual CSVD reconstructions with the weights produced by VCRN. The VCRN is trained as an image classifier to produce corruption class prediction probabilities, *s.t.*,

$$\text{VCRN}(\mathbf{i}) = [p_1, \cdots, p_{n_I}] \in [0, 1]^{n_I} \quad (8)$$

where n_I is the number of corruption classes. The top-k probabilities in descending order are filtered, and the image is routed to the CSVD modules of the associated corruption classes. We define $\text{CSVD}_{c_I} : c_I(\mathcal{I}_S) \rightarrow \mathcal{I}_S$, as an image reconstruction network trained on clean and corrupted image pairs. The resultant image is produced as the weighted sum of the CSVD deconstruction:

$$\hat{\mathbf{i}} = W(\mathbf{i}) \cdot \sum_{j=1}^{k} \text{CSVD}_j(\mathbf{i}) \quad (9)$$

While there can be multiple ways to generate $W(\cdot)$ [14, 99], we use the softmax over top-k logits similar to [44, 71] but without a trainable linear layer, *i.e.*,

$$W(\mathbf{i}) = \text{Softmax}(\text{Top-}k(\text{VCRN}(\mathbf{i}))). \quad (10)$$

4.3. Textual DeNoiser (TDN) Module

The TDN is a generative language model prompted to denoise, either using the generative VQA model g_θ itself for self-denoising or a separate model. Regardless of the type of TDN used, we denote it as d_T, s.t.,

$$d_T : \bigcup_{c_T\in\mathcal{C}_T} c_T(\mathcal{Q}) \rightarrow \mathcal{Q}. \quad (11)$$

The prompted TDN can be defined as per Eq. (2) but without the visual tokens. We define a textual denoising prompting function $\mathcal{P}(\cdot)$ to form the denoised textual tokens $d_T(\mathbf{q}) =< t_1, t_2, ...t_n >$ as,

$$t_i = \arg\max \mathbb{P}(t|\mathcal{P}(c_T(\mathbf{q})), t_{<i}). \quad (12)$$

It should be noted that the mapping in Eq. (11) differs slightly from Eq. (5). While a TDN and VDN can have similar cross-distribution formulations, it is hard to define the source distribution of LLMs as these models are typically pre-trained on the entire internet [21].

5. Vision Language System Robustness Benchmark (VLSRB)

VLSRB extends the VRB [40] to the textual modality by introducing 18 realistic textual corruptions, adding 4 new visual corruptions, and integrating denoisers in

each modality as preprocessors.

Visual Corruptions. We expand VRB's [40] 14 visual corruptions by adding four new ones from ImageNet-C [32], and Imgaug [46]. The functions have been categorized into 6 corruption classes: Additive Noise, Digital, Image Attribute Transformation, Weather, Blur, and Physical (Fig. 12). As visual corruptions can have multiple severity levels [32], we randomly select a single severity level for each sample while augmenting the dataset.

Textual Corruptions. Following NLPaug [59] and MMCBench [92], we select 18 textual corruption functions to replicate character, word, and sentence-level perturbations in realistic scenarios (Tab. 8). Unlike its visual counterparts, the textual corruptions have a single severity level.

Denoisers. The visual denoiser consists of two parts: a noise router and a set of corruption-specific denoisers (§4.2). Both modules are trained on the source distribution (§6). For routing, we opt to fine-tune a ResNet-50 [31] on 18 corruption classes, achieving a validation accuracy of 98.16%, and for corruption-specific denoising, we use DRUNet [41]. We handle textual denoising using a frozen LLM in zero-shot settings to reconstruct the corrupted text (§4.3), with Gemini 2.0 Flash serving as our choice of the LLM using the prompt below. Detailed ablations on the denoisers are provided in §8.1, 8.2.

Zero-Shot Prompt for Textual Denoising

You are an expert at denoising text. Your task is to provide the denoised version of a given noisy question. Follow these instructions:
1. Return only the denoised version of the text or question.
2. Do not provide explanations or additional words.
3. Do not answer the question or alter its intent.
4. Maintain the question format if the input is a question.
5. Avoid presenting the answer in assertive form.
You will receive a noisy text or a question. Provide the denoised version as specified.

Question: `<QUESTION>`

6. Experimental Setup

Tasks & Datasets. We chose Visual Question Answering (VQA) as our evaluation task, primarily due to its versatility and the high-level understanding of the vision-language modalities it requires. For the source and target distributions, we used the VQAv2 [30] and DARE [74] datasets, respectively. The DARE dataset has been selected as (a) it contains a good variety of VQA sub-tasks, *e.g.*, count, order, trick, and visual commonsense reasoning for extensive textual evaluation, (b) the visual distribution differs significantly from the source image distribution of VQAv2 (Tab. 1).

Dataset Splits	**FID** [35]	**GS** [47]	**MSID** [77]
Intra-VQAv2.0 [30]	22.34	000.71	00.35
VQAv2.0 *vs.* DARE [74]	76.28	797.42	49.67

Table 1. Distribution shift scores between datasets. Intra-dataset shift is measured by taking two mutually exclusive random subsets. The high score differences indicate a significant distribution shift between the VQAv2.0 and DARE.

Visual Denoiser Training Dataset. We construct an augmented image dataset, similar to VRB [40], by randomly sampling 3000 images from the source distribution, VQAv2's validation set, and augmenting them using our 18 visual corruptions at 5 severity levels. The resulting dataset contains 3000 x 18 x 5 = 270,000 images and has a 90:10 train-val split. The training split of the whole dataset is used to train the corruption classifier, and the individual corruption subsets are used to train each denoiser independently. The training and implementation details are provided in §11.

VLM Baselines. The VLMs have been evaluated on the 1 correct subset of the DARE dataset [74]. We evaluated four open-source and one closed-source model, as per Tab. 5. The models have been prompted under the zero-shot setting using DARE's default prompt (§9).

Human Baseline. To evaluate whether the corrupted samples are still interpretable, we establish a human performance baseline. We randomly sample 2000 image-question pairs from the corrupted subset of each category in the DARE dataset. The pairs are presented to five human annotators, who attempt to answer the corrupted input. The annotators were provided monetary compensation for their work. The human baseline establishes a reference point for evaluating the system's robustness and answerability of the questions (§16).

7. Result Analysis

Robustness Enhancement. As presented in Tab. 2, VLMs experience a 3% to 10% overall performance drop when subjected to corruption in both modalities. Introducing visual, textual, or both denoisers tends to improve performance in most cases, reaching up to 9% for certain models. On average, we observe a consistent

Model	Approach	Accuracy				Overall
		Count	Order	Trick	VCR	
LLaVA-v1.6 7B	Clean	40.40	39.39	54.74	47.20	45.37
	Corr	$28.52_{\pm 3.16}$	$32.48_{\pm 3.36}$	$43.88_{\pm 3.89}$	$40.85_{\pm 2.21}$	$36.35_{\pm 6.20}$
	Corr + VDN	$29.63_{\pm 4.27}(+1.11)$	$32.09_{\pm 3.23}(-0.39)$	$42.10_{\pm 3.09}(-1.78)$	$41.49_{\pm 2.22}(+0.64)$	$36.30_{\pm 5.56}(-0.05)$
	Corr + TDN	$35.73_{\pm 1.79}(+7.21)$	$35.68_{\pm 1.89}(+3.20)$	$49.04_{\pm 2.29}(+5.16)$	$42.50_{\pm 1.26}(+1.65)$	$40.68_{\pm 5.48}(+4.33)$
	Corr + VDN + TDN	$37.41_{\pm 1.71}(+8.89)$	$34.76_{\pm 1.80}(+2.28)$	$46.53_{\pm 1.87}(+2.65)$	$43.01_{\pm 1.43}(+2.16)$	$40.42_{\pm 4.56}(+4.07)$
InstructBLIP 7B	Clean	32.40	43.72	50.00	52.80	44.65
	Corr	$30.86_{\pm 2.56}$	$37.94_{\pm 3.20}$	$46.78_{\pm 2.62}$	$49.67_{\pm 2.26}$	$41.28_{\pm 7.51}$
	Corr + VDN	$30.88_{\pm 2.52}(+0.02)$	$38.49_{\pm 3.22}(+0.55)$	$46.66_{\pm 2.82}(-0.12)$	$49.48_{\pm 2.12}(-0.19)$	$41.33_{\pm 7.37}(+0.05)$
	Corr + TDN	$32.53_{\pm 0.97}(+1.67)$	$41.59_{\pm 1.30}(+3.65)$	$49.68_{\pm 1.69}(+2.90)$	$51.79_{\pm 1.74}(+2.12)$	$43.83_{\pm 7.68}(+2.55)$
	Corr + VDN + TDN	$32.04_{\pm 0.90}(+1.18)$	$42.08_{\pm 1.33}(+4.14)$	$49.69_{\pm 1.67}(+2.91)$	$51.50_{\pm 1.73}(+1.83)$	$43.75_{\pm 7.77}(+2.47)$
Janus-Pro 7B	Clean	48.40	65.37	70.26	60.80	60.96
	Corr	$36.77_{\pm 4.56}$	$52.35_{\pm 7.07}$	$58.98_{\pm 3.81}$	$53.40_{\pm 2.06}$	$50.17_{\pm 8.32}$
	Corr + VDN	$38.36_{\pm 3.93}(+1.59)$	$52.80_{\pm 7.23}(+0.45)$	$59.70_{\pm 3.99}(+0.72)$	$54.24_{\pm 1.89}(+0.84)$	$51.09_{\pm 7.95}(+0.92)$
	Corr + TDN	$44.16_{\pm 2.30}(+7.39)$	$59.98_{\pm 4.15}(+7.63)$	$62.30_{\pm 2.91}(+3.32)$	$54.69_{\pm 1.31}(+1.29)$	$55.06_{\pm 7.02}(+4.89)$
	Corr + VDN + TDN	$44.60_{\pm 2.68}(+7.83)$	$60.14_{\pm 4.45}(+7.79)$	$63.70_{\pm 3.32}(+4.72)$	$54.82_{\pm 1.31}(+1.42)$	$55.58_{\pm 7.23}(+5.41)$
Idefics2 8B	Clean	56.00	52.38	64.66	60.80	58.46
	Corr	$42.78_{\pm 6.29}$	$47.25_{\pm 2.87}$	$58.01_{\pm 2.71}$	$57.17_{\pm 1.50}$	$51.26_{\pm 6.52}$
	Corr + VDN	$45.20_{\pm 5.17}(+2.42)$	$47.42_{\pm 3.18}(+0.17)$	$58.82_{\pm 2.65}(+0.81)$	$57.85_{\pm 1.46}(+0.68)$	$52.30_{\pm 6.08}(+1.04)$
	Corr + TDN	$49.43_{\pm 3.30}(+6.65)$	$48.66_{\pm 1.65}(+1.41)$	$60.25_{\pm 1.76}(+2.24)$	$57.75_{\pm 1.33}(+0.58)$	$54.01_{\pm 5.03}(+2.75)$
	Corr + VDN + TDN	$50.77_{\pm 3.53}(+7.99)$	$49.09_{\pm 1.63}(+1.84)$	$61.64_{\pm 1.95}(+3.63)$	$58.92_{\pm 1.52}(+1.75)$	$55.10_{\pm 5.25}(+3.84)$
Gemini-2.0 Flash	Clean	69.60	64.50	79.31	65.60	69.67
	Corr	$50.86_{\pm 6.07}$	$56.21_{\pm 5.55}$	$68.08_{\pm 3.52}$	$62.91_{\pm 2.10}$	$59.42_{\pm 6.54}$
	Corr + VDN	$49.86_{\pm 6.28}(-1.00)$	$55.83_{\pm 4.10}(-0.38)$	$67.90_{\pm 3.48}(-0.18)$	$61.11_{\pm 1.91}(-1.80)$	$58.56_{\pm 6.64}(-0.86)$
	Corr + TDN	$52.30_{\pm 5.61}(+1.44)$	$57.93_{\pm 5.94}(+1.72)$	$68.10_{\pm 4.51}(+0.02)$	$63.49_{\pm 2.04}(+0.58)$	$60.36_{\pm 5.94}(+0.94)$
	Corr + VDN + TDN	$51.63_{\pm 5.66}(+0.77)$	$56.74_{\pm 5.94}(+0.53)$	$67.96_{\pm 4.94}(-0.12)$	$61.61_{\pm 1.84}(+1.30)$	$59.38_{\pm 6.02}(-0.04)$
Model Average	Clean	49.36	53.07	63.79	57.44	55.82
	Corr	$37.96_{\pm 10.11}$	$45.25_{\pm 10.42}$	$55.15_{\pm 9.89}$	$52.80_{\pm 8.12}$	$47.70_{\pm 6.80}$
	Corr + VDN	$38.45_{\pm 9.40}(+0.49)$	$45.27_{\pm 10.14}(+0.02)$	$54.67_{\pm 10.32}(-0.47)$	$52.64_{\pm 7.62}(-0.16)$	$47.67_{\pm 6.45}(-0.03)$
	Corr + TDN	$42.83_{\pm 9.32}(+4.87)$	$48.77_{\pm 10.49}(+3.52)$	$57.87_{\pm 8.63}(+2.73)$	$54.04_{\pm 7.56}(+1.24)$	$50.78_{\pm 5.68}(+3.08)$
	Corr + VDN + TDN	$43.09_{\pm 8.73}(+5.13)$	$48.56_{\pm 10.21}(+3.31)$	$57.61_{\pm 9.33}(+2.46)$	$53.84_{\pm 7.24}(+1.04)$	$50.69_{\pm 5.49}(+2.99)$
Human	Corr	80.90	77.15	78.45	81.05	79.40

Table 2. Evaluation of model robustness using different denoising methods. *Clean* denotes the baseline performance of the models for uncorrupted input, and *Corr* denotes the performance for corrupted input. The mean accuracy across noise types, their standard deviations, and the mean difference with the corrupted baseline have been reported. *VDN* and *TDN* refer to the visual and textual denoisers respectively.

3% accuracy bump across models, with Janus-Pro 7B gaining a 5.5% increase when using both denoisers.

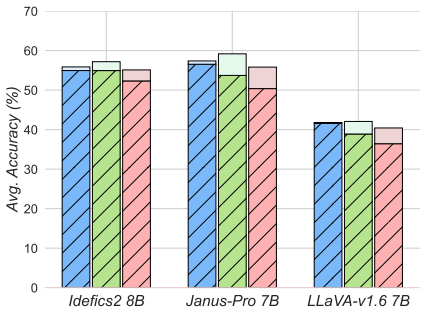
Performance Drops. We observe in Tab. 2 several instances of performance drop when incorporating the visual denoiser. Notably, Gemini 2.0 Flash experiences consistent degradation across all categories, suggesting that visual denoising isn't yet a generalized tool applicable to all VLMs. A deeper error analysis identifies the pitfall of visual denoising §8.3.

Visual *vs*. Textual Denoising. From Tab. 2, it is evident that textual denoising performs noticeably better than visual denoising. Averaging across models, textual denoising yields a modest 3% performance gain, while visual denoising drops accuracy slightly for all instances, except a few cases, *e.g.*, Idefics2 in VCR. This discrepancy can be attributed to either the weakness of the visual denoiser (§8.3) or textual perturbations (§9.1).

Robustness across Corruption Types. From Fig. 4, our visual denoising framework shows significant improvements on *zoom blur* and *rain* effects, but struggles with *elastic* noise. This weakness can be attributed to the excessive blurring introduced while denoising elastic noise (Figs. 18 and 21). The textual denoisers perform relatively better across all perturbations, with the highest gains against *character insertion*, and minimal improvements for *repetition*, *substitution*, and *paraphrasing* perturbations. Performance variations of models across corruption types are shown in Fig. 17.

Can we fully remove noise? Following Fig. 3a, models such as InstructBLIP-7B [19] nearly recover their clean accuracy when using both visual and textual denoisers, while Gemini 2.0 Flash has a substantial gap between denoised and uncorrupted performance. We also observe some corruptions being harder to denoise, *e.g.*, in the visual modality, *rain* and *elastic* noise severely degrade image quality (Figs. 18 and 19), and in

(a) Average accuracy gap between clean and denoised inputs, using both denoisers, for visual (top) & textual (bottom) corruptions.



(b) Effect of applying corruption followed by ***modality-specific denoising*** on three VLMs. For each model, the three bars represent (from left): (1) corrupting only the ***image modality*** and then denoising it (question kept clean), (2) corrupting only the ***question*** and then denoising it (image kept clean), and (3) corrupting both modalities and denoising them. Shaded regions denote *Corrupted*, Unshaded regions denote *Denoised*.

Figure 3

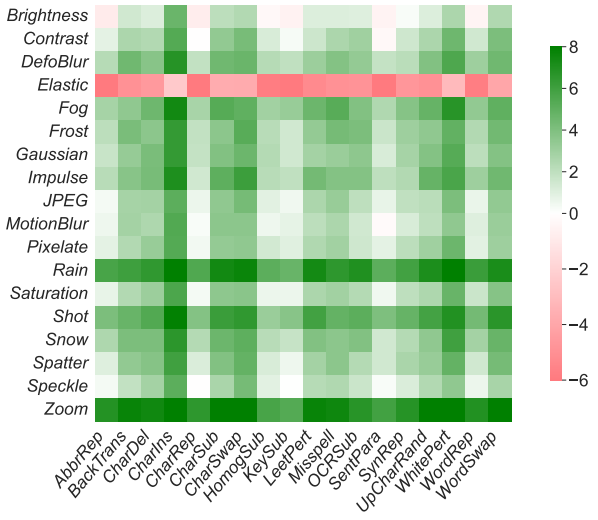


Figure 4. The x and y axes represent the textual and visual corruptions, respectively. Each cell denotes the accuracy difference between denoised and corrupted inputs, *i.e.*, between *Corr+VDN+TDN* and *Corr*.

the textual modality, *homoglyph substitution*, *synonym replacement*, and *character deletion* often make it harder to reconstruct the original sentence (Tab. 8).

Which modality matters more? In Fig. 3b, we corrupt and denoise one modality while keeping the other one clean. Denoising text or both modalities yields the highest performance gains, whereas visual-only denoising produces marginal improvements. While this might suggest that textual information plays a more critical role in corruption robustness, there can be more nuances to this claim (§9.1).

Parameter Overhead. Both the visual and textual denoisers produce minimal parameter overhead. The routing module of the visual denoiser uses ResNet-50 with a parameter count of 25.6M, and the denoiser DRUNet has a parameter count of 32.64M. The number of denoisers matches the number of visual corruptions, *i.e.*, 18 denoisers are called. Hence, the total active parameter count is $25.6+32.64*18 = 613.12$M, roughly 8.05% of a standard 7B VLM system.

The parameter count of the textual denoiser is harder to determine, as we are using Gemini-2.0 Flash, a closed-source model. However, the inference time is fast, with one of the cheapest API costs at only \$0.10 for 1M tokens[1]. Textual denoisers also don't have any training overhead like their visual counterparts. Small language models can be a good alternative, with a parameter count as low as 1B [24].

8. Ablations and Qualitative Analyses

Additional analysis is provided in §15.

8.1. Ablation on Visual Denoisers

We train DnCNN [93], BRDNet [76], and DRUNet [41] using the MSE loss against the clean image. The models are evaluated using two categories of metrics:

[1]https://ai.google.dev/gemini-api/docs/pricing#gemini-2.0-flash

Model	BLEU ↑	METEOR ↑	R-1 ↑	R-2 ↑	R-L ↑	WER ↓	BERTScore ↑	S-BERT ↑
GPT-4o	0.757	0.902	0.912	0.845	0.909	0.211	0.946	0.915
Gemini-2.0 Flash	**0.767**	**0.904**	**0.915**	**0.851**	**0.913**	**0.201**	**0.948**	**0.917**
DeepSeek LLM 7B Chat	0.578	0.772	0.787	0.634	0.768	0.397	0.859	0.777

Table 3. Model performance comparison across evaluation metrics. **BLEU** [64], **METEOR** [3], and **ROUGE** [56] assess lexical similarity, while **WER** measures transcription accuracy (lower is better). **BERTScore** [97] and **Sentence-BERT** [68] capture semantic similarity. **Bold** values indicate the best model. ↑ denotes higher is better, and ↓ indicates lower is preferable.

Full Reference (PSNR, SSIM [83], LPIPS [95]) and No Reference (CLIP_IQA [81], BRISQUE [61]). Higher PSNR/SSIM and lower LPIPS indicate better perceptual quality, while higher CLIP_IQA and lower BRISQUE reflect improved visual alignment and fewer distortions. Following Tab. 4, we found DRUNet achieving the best overall performance and have selected it as our method for denoising. The performance on individual visual corruptions has been reported in Tab. 9.

Method	Full Reference Metrics			No Reference Metrics	
	PSNR ↑	SSIM ↑	LPIPS ↓	CLIP ↑	BRISQUE ↓
DnCNN	25.17	0.78	0.16	0.65	24.54
BRDNet	25.39	0.79	0.15	0.66	24.01
DRUNet	**27.88**	**0.85**	**0.09**	**0.76**	**23.07**

Table 4. Performance comparison of models across evaluation metrics. Metrics are categorized into **Full Reference** (PSNR, SSIM [83], LPIPS [95]) and **No Reference** (CLIP_IQA [81], BRISQUE [61]). **Bold** values denote the best performance. ↑ indicates higher is better, and ↓ means lower is preferable.

8.2. Ablation on Textual Denoisers

We evaluate three LLMs: GPT-4o [1], Gemini-2.0 Flash [29], and DeepSeek-LLM-7B-Chat [21], as textual denoisers. We assess the denoised text on lexical similarity using BLEU [64], METEOR [3], and ROUGE [56] scores, transcription accuracy using WER, and semantic similarity using BERTScore [97] and Sentence-BERT [68]. We select Gemini-2.0 Flash as the textual denoiser for our benchmark, as it achieves the best overall performance (Tab. 3).

8.3. Qualitative Analysis of Visual Denoising

Given the marginal quantitative gains of visual denoising, it is natural to question the efficacy of visual denoising. To investigate this, we examine the visual explainability of the denoised outputs using the attention distributions of CAM-based methods (Fig. 7). We find that denoised images partially restore the attention maps of their clean counterparts, indicating genuine recovery of salient visual features. We also observe the robustness of models across severity levels (Fig. 6).

However, there are several cases where the visual denoiser fails. As shown in Fig. 21, denoising sometimes introduces excessive blurring or artifacts, especially when applied to complex corruptions such as *elastic* noise. Similar, though less severe, effects appear against other corruption classes (Figs. 18 to 20). These artifacts can inadvertently make the images more adversarial, which might explain the instances where performance drops after visual denoising.

Figure 6. The original image progressively corrupted by *Contrast* noise with *(severity 1–5)* in the first row and the corresponding denoised images using DRUNet [41] in the second row, exemplifying robustness across severity levels.

8.4. Qualitative Analysis of Textual Denoising

Textual denoising performs well both quantitatively (Tab. 2) and qualitatively (Tab. 8). Textual corruptions often disrupt image–text alignment by distorting the attention map, which is mostly restored after textual denoising (Fig. 5). Nonetheless, the textual denoiser fails in cases where the word replacements or paraphrases are linguistically correct but semantically inconsistent with the image. *E.g.*, in Tab. 8, "lego figures" is replaced with "lego statues", which doesn't align with the image and hence, has been incorrectly denoised.

9. Discussion

9.1. Are textual perturbations weak?

Our results showed that textual denoising yielded noticeably better results, which can be attributed to the textual bias of VLMs [30]. However, one can also argue the weakness of the textual corruption strength. Visual corruptions are applied at five severity levels, whereas textual perturbations exist only at a single level. Stronger textual perturbations would likely induce greater degradation and challenge the denoiser. Therefore, we cau-

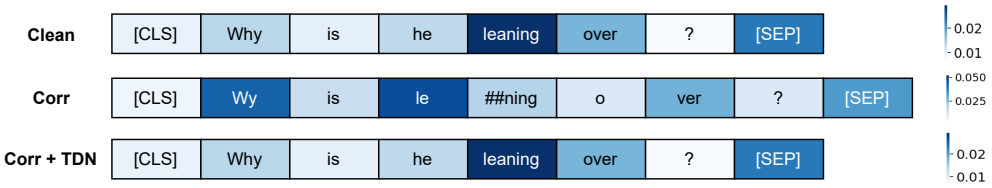


Figure 5. Attention maps of an input question under clean, corrupted, and denoised conditions, generated by feeding an image-question pair into ViLT[48]. We observe the disruption and restoration of attention maps via denoising.

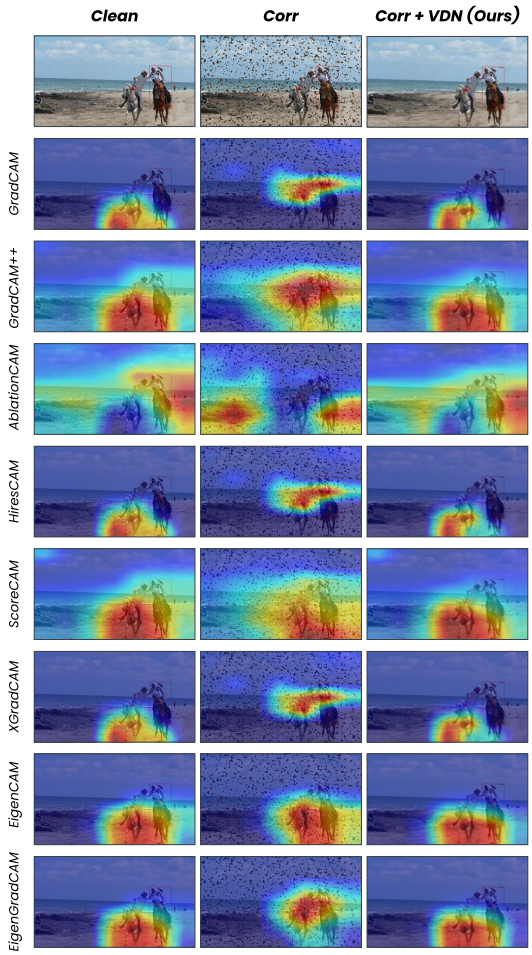


Figure 7. Visual explainability of our denoising framework on the Splatter noise. Each set shows clean, corrupted, and denoised images with corresponding CAM-based explanations [7, 22, 23, 26, 63, 63, 70, 80].

tion against concluding that textual denoisers should be prioritized over visual ones.

9.2. Model *vs*. System Robustness

Measuring system robustness requires isolating the contribution of denoisers from the model's inherent robustness. For instance, if a model is naturally resistant to a particular corruption, both the performance drop under corruption and the subsequent denoising gains will appear marginal. This may explain why visual denoisers yield smaller improvements, as the VLMs could already possess stronger robustness in the visual modality. A possible form of evaluation would be to introduce a metric that accounts for both the performance drop due to corruption and the denoising performance gains.

9.3. Out-of-Domain (OOD) Corruption

Our current evaluation assumes the corruption classes at test time are known, but in practice, models will inevitably encounter unseen corruptions. Our textual denoisers using LLMs are naturally more robust to such shifts, as they are (i) inherently used in a training-free/zero-shot setting, and (ii) pre-trained on a large corpus of noisy text, including several forms of realistic textual perturbations.

However, our visual denoiser relies on corruption-specific or corruption-aware training. Consequently, unseen corruptions are routed to the closest known denoiser, which can potentially degrade image quality or introduce artifacts. One probable mitigation strategy is to train the router with an additional denoiser class for unseen corruptions. During inference, these unseen, corrupted inputs could either bypass the denoiser or be routed through a generalized denoiser.

10. Conclusion

We propose a denoising framework to enhance the robustness of Vision-Language Models (VLMs) against real-world multimodal corruptions. Our comprehensive experiments assess the sensitivity of VLMs to visual and textual perturbations across different types of VQA tasks. We present key findings on denoising, robustness against specific corruptions, and the importance of each modality. By identifying modality-specific vulnerabilities and denoising effectiveness, our study offers key insights for enhancing VLM robustness and reliability and opens new avenues for future VLM research.

Acknowledgments: We sincerely appreciate Ishmam Tashdeed and Reaz Hassan Joarder for their guidance.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 7

[2] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. Vqa: Visual question answering, 2016. 1

[3] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, 2005. Association for Computational Linguistics. 7

[4] Olivia Brown and Brad Dillman. Proceedings of the robust artificial intelligence system assurance (raisa) workshop 2022. *arXiv preprint arXiv:2202.04787*, 2022. 1

[5] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005. 2

[6] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness, 2019. 2

[7] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018. 8

[8] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark, 2018. 2

[9] Hanjie Chen and Yangfeng Ji. Adversarial training for improving model robustness? look at both prediction and interpretation, 2022. 2

[10] Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. Counterfactual samples synthesizing for robust visual question answering, 2020. 2

[11] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration, 2022. 2

[12] Shuo Chen, Jindong Gu, Zhen Han, Yunpu Ma, Philip Torr, and Volker Tresp. Benchmarking robustness of adaptation methods on pre-trained vision-language models. *Advances in Neural Information Processing Systems*, 36:51758–51777, 2023. 2

[13] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling, 2025. 2

[14] Aidan Clark, Diego de Las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, et al. Unified scaling laws for routed language models. In *International conference on machine learning*, pages 4057–4086. PMLR, 2022. 3

[15] Jeremy M Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing, 2019. 2

[16] Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, Yang Zhou, Kaizhao Liang, Jintai Chen, Juanwu Lu, Zichong Yang, Kuei-Da Liao, et al. A survey on multimodal large language models for autonomous driving. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 958–979, 2024. 1

[17] Xuanming Cui, Alejandro Aparcedo, Young Kyun Jang, and Ser-Nam Lim. On the robustness of large multimodal models against image adversarial attacks, 2023. 1

[18] Yuning Cui, Syed Waqas Zamir, Salman Khan, Alois Knoll, Mubarak Shah, and Fahad Shahbaz Khan. Adair: Adaptive all-in-one image restoration via frequency mining and modulation, 2024. 2

[19] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. 5, 2

[20] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. Visual dialog. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 326–335, 2017. 1

[21] DeepSeek-AI. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024. 3, 7

[22] Saurabh Desai and Harish G. Ramaswamy. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 972–980, 2020. 8

[23] Rachel Lea Draelos and Lawrence Carin. Use hirescam instead of grad-cam for faithful explanations of convolutional neural networks, 2021. 8

[24] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024. 6

[25] William Fedus, Jeff Dean, and Barret Zoph. A review of sparse expert models in deep learning. *arXiv preprint arXiv:2209.01667*, 2022. 1, 3

[26] Ruigang Fu, Qingyong Hu, Xiaohu Dong, Yulan Guo, Yinghui Gao, and Biao Li. Axiom-based grad-cam: Towards accurate visualization and explanation of cnns, 2020. 8

[27] Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, and Lingpeng Kong. G-llava: Solving geometric problem with multi-modal large language model, 2023. 1

[28] Lovedeep Gondara. Medical image denoising using convolutional denoising autoencoders. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 241–246, 2016. 2

[29] Google DeepMind. Google Gemini AI update, december 2024. https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/, 2024. Accessed: 27-03-25. 7, 2

[30] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4, 7

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 4

[32] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations, 2019. 4

[33] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations, 2019. 3

[34] Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. Pretrained transformers improve out-of-distribution robustness, 2020. 2

[35] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. 4

[36] Jia-Hong Huang, Cuong Duc Dao, Modar Alfadly, and Bernard Ghanem. A novel framework for robustness analysis of visual qa models, 2018. 2

[37] Jia-Hong Huang, Cuong Duc Dao, Modar Alfadly, and Bernard Ghanem. A novel framework for robustness analysis of visual qa models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8449–8456, 2019. 1

[38] Zhuo Huang, Chang Liu, Yinpeng Dong, Hang Su, Shibao Zheng, and Tongliang Liu. Machine vision therapy: Multimodal large language models can enhance visual robustness via denoising in-context learning. *arXiv preprint arXiv:2312.02546*, 2023. 2

[39] Md Farhan Ishmam, Md Sakib Hossain Shovon, Muhammad Firoz Mridha, and Nilanjan Dey. From image to language: A critical analysis of visual question answering (vqa) approaches, challenges, and opportunities. *Information Fusion*, 106:102270, 2024. 1

[40] Md Farhan Ishmam, Ishmam Tashdeed, Talukder Asir Saadat, Md Hamjajul Ashmafee, Abu Raihan Mostofa Kamal, and Md. Azam Hossain. Visual robustness benchmark for visual question answering (vqa), 2024. 1, 2, 3, 4

[41] Mina Jafari, Dorothee Auer, Susan Francis, Jonathan Garibaldi, and Xin Chen. Dru-net: An efficient deep convolutional neural network for medical image segmentation, 2020. 2, 4, 6, 7

[42] Erica Jen. Stable or robust? what's the difference. *Robust Design: a repertoire of biological, ecological, and engineering case studies*, pages 7–20, 2005. 1, 2

[43] Jiabao Ji, Bairu Hou, Zhen Zhang, Guanhua Zhang, Wenqi Fan, Qing Li, Yang Zhang, Gaowen Liu, Sijia Liu, and Shiyu Chang. Advancing the robustness of large language models through self-denoised smoothing, 2024. 2

[44] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024. 3

[45] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution, 2016. 1

[46] Alexander B. Jung, Kentaro Wada, Jon Crall, Satoshi Tanaka, Jake Graving, Christoph Reinders, Sarthak Yadav, Joy Banerjee, Gábor Vecsei, Adam Kraft, Zheng Rui, Jirka Borovec, Christian Vallentin, Semen Zhydenko, Kilian Pfeiffer, Ben Cook, Ismael Fernández, François-Michel De Rainville, Chi-Hung Weng, Abner Ayala-Acevedo, Raphael Meudec, Matias Laporte, et al. imgaug. https://github.com/aleju/imgaug, 2020. Online; accessed 01-Feb-2025. 4

[47] Valentin Khrulkov and Ivan Oseledets. Geometry score: A method for comparing generative adversarial networks, 2018. 4

[48] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision, 2021. 8

[49] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 1

[50] Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? *Advances in Neural Information Processing Systems*, 37:87874–87907, 2024. 1

[51] Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models?, 2024. 2

[52] Boyun Li, Xiao Liu, Peng Hu, Zhongqin Wu, Jiancheng Lv, and Xi Peng. All-in-one image restoration for unknown corruption. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17431–17441, 2022. 2

[53] Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day, 2023. 1

[54] Chunyi Li, Jianbo Zhang, Zicheng Zhang, Haoning Wu, Yuan Tian, Wei Sun, Guo Lu, Xiaohong Liu, Xiongkuo Min, Weisi Lin, and Guangtao Zhai. R-bench: Are your large multimodal model robust to real-world corruptions?, 2024. 1

[55] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer, 2021. 2

[56] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, 2004. Association for Computational Linguistics. 7

[57] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2024. 2

[58] Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation, 2020. 2

[59] Edward Ma. Nlp augmentation. https://github.com/makcedward/nlpaug, 2019. 4

[60] Eric Mintun, Alexander Kirillov, and Saining Xie. On interaction between augmentations and corruptions in natural corruption robustness. In *Advances in Neural Information Processing Systems*, pages 3571–3583. Curran Associates, Inc., 2021. 2

[61] Anish Mittal, Anush K. Moorthy, and Alan C. Bovik. Blind/referenceless image spatial quality evaluator. In *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pages 723–727, 2011. 7

[62] Sameera V. Mohd Sagheer and Sudhish N. George. A review on medical image denoising algorithms. *Biomedical Signal Processing and Control*, 61:102036, 2020. 2

[63] Mohammed Bany Muhammad and Mohammed Yeasin. Eigen-cam: Class activation map using principal components. In *2020 International Joint Conference on Neural Networks (IJCNN)*, page 1–7. IEEE, 2020. 8

[64] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, 2002. Association for Computational Linguistics. 7

[65] 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, 2024. 2

[66] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 1

[67] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A. Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Data augmentation can improve robustness, 2021. 2

[68] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2019. 7

[69] Christophe Rigaud, Antoine Doucet, Mickaël Coustaty, and Jean-Philippe Moreux. Icdar 2019 competition on post-ocr text correction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1588–1593, 2019. 2

[70] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, 2019. 8

[71] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017. 3

[72] Eric Silverman and Takashi Ikegami. Robustness in artificial life. *International Journal of Bio-Inspired Computation 9*, 3(3):179–186, 2011. 1, 2

[73] Sarvesh Soni, Meghana Gudala, Atieh Pajouhi, and Kirk Roberts. RadQA: A question answering dataset to improve comprehension of radiology reports. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6250–6259, Marseille, France, 2022. European Language Resources Association. 1

[74] Hannah Sterz, Jonas Pfeiffer, and Ivan Vulić. Dare: Diverse visual question answering with robustness evaluation, 2024. 4, 3

[75] Yifu Sun and Haoming Jiang. Contextual text denoising with masked language models, 2024. 2

[76] Chunwei Tian, Yong Xu, and Wangmeng Zuo. Image denoising using deep cnn with batch renormalization. *Neural Networks*, 121:461–473, 2020. 6

[77] Anton Tsitsulin, Marina Munkhoeva, Davide Mottin, Panagiotis Karras, Alex Bronstein, Ivan Oseledets, and Emmanuel Müller. The shape of data: Intrinsic distance for data distributions, 2020. 4

[78] Weijie Tu, Weijian Deng, and Tom Gedeon. Toward a holistic evaluation of robustness in clip models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 2

[79] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator, 2015. 1

[80] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks, 2020. 8

[81] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *AAAI*, 2023. 7

[82] Liang Wang, Wei Zhao, Ruoyu Jia, Sujian Li, and Jingming Liu. Denoising based sequence-to-sequence pre-training for text generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4003–4015, Hong Kong, China, 2019. Association for Computational Linguistics. 2

[83] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 7

[84] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, 2020. Association for Computational Linguistics. 1

[85] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention, 2016. 1

[86] Yihao Xue, Kyle Whitecross, and Baharan Mirzasoleiman. Investigating why contrastive learning benefits robustness against label noise, 2022. 2

[87] Li Yang, Yan Xu, Chunfeng Yuan, Wei Liu, Bing Li, and Weiming Hu. Improving visual grounding with visual-linguistic verification and iterative reasoning, 2022. 1

[88] Songhyun Yu, Bumjun Park, and Jechang Jeong. Deep iterative down-up cnn for image denoising. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2095–2103, 2019. 2

[89] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration, 2022. 2

[90] Jiehang Zeng, Xiaoqing Zheng, Jianhan Xu, Linyang Li, Liping Yuan, and Xuanjing Huang. Certified robustness to text adversarial attacks by randomized [mask], 2021. 2

[91] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5625–5644, 2024. 1

[92] Jiawei Zhang, Tianyu Pang, Chao Du, Yi Ren, Bo Li, and Min Lin. Benchmarking large multimodal models against common corruptions, 2024. 1, 2, 4

[93] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. 2, 6

[94] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27 (9):4608–4622, 2018. 2

[95] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *CoRR*, abs/1801.03924, 2018. 7

[96] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018. 1

[97] Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020. 7

[98] Zhen Zhang, Guanhua Zhang, Bairu Hou, Wenqi Fan, Qing Li, Sijia Liu, Yang Zhang, and Shiyu Chang. Certified robustness for large language models with self-denoising. *arXiv preprint arXiv:2307.07171*, 2023. 2

[99] Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022. 3

[100] Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. Freelb: Enhanced adversarial training for natural language understanding, 2020. 2

Enhancing Vision Language Corruption Robustness using Cross-Distribution & Prompted Denoisers

Supplementary Material

11. Training Details

The experiments were conducted using NVIDIA RTX 3090 24GB GPUs, utilizing the Hugging Face implementation of the models [84]. We fine-tuned each visual denoiser for 50 epochs with a learning rate of 0.0001 and a batch size of 30. We use Mean Squared Error (MSE) as our loss function and Adam optimizer [49] for faster convergence. Before training, all the images were resized to a dimension of 224x224. Early stopping was used to prevent overfitting. The ResNet-50 requires 3h 31m training time, and the training time of the denoisers for each corruption has been reported in Tab. 6.

Corruption	Time	Corruption	Time
Brightness	5h 49m	Saturation	5h 43m
Contrast	4h 43m	Speckle	2h 26m
Defocus Blur	4h 24m	Zoom Blur	4h 32m
Elastic	1h 29m	Motion	4h 25m
Fog	4h 49m	Pixelate	1h 44m
Frost	4h 44m	Rain	3h 29m
Gaussian	2h 44m	Shot	5h 27m
JPEG	4h 29m	Snow	4h 59m
Impulse	3h 21m	Spatter	3h 12m

Table 6. Total Training Time for Different Corruptions

12. Ablation on Loss Function

We ablate on our loss function by using other forms of loss, LPIPS [96] and VGG Loss [45]. We found mixed results with MSE outperforming in terms of PSNR and SSIM, and VGG in terms of LPIPS, both indicating better perceptual quality. LPIPS loss surprisingly does worse than VGG Loss in its own metric but outperforms the rest in CLIP_IQA and BRISQUE, indicating great alignment of the visual content.

Model	Full Reference Metrics			No Reference Metrics	
	PSNR ↑	SSIM ↑	LPIPS ↓	CLIP_IQA ↑	BRISQUE ↓
MSE	**26.72**	**0.84**	0.1108	0.742	23.603
LPIPS	25.36	0.83	0.0899	**0.778**	**20.787**
VGG	25.93	0.83	**0.0897**	0.768	26.286

Table 7. Denoising evaluation of our best denoiser DRUNet trained with different loss functions. ↑ indicates higher is better, and ↓ means lower is preferable.

13. Examples of Perturbations

13.1. Examples of Textual Perturbations

The details of the 18 textual perturbations applied to questions in our experiments are categorized in Tab. 8 into word-level, sentence-level, and character-level perturbations. These perturbations mimic real-world challenges like typos, abbreviations, synonym replacements, and OCR errors, testing model robustness against imperfect inputs and linguistic variations.

13.2. Examples of Image Perturbations

Examples of 18 visual corruptions used in our experiments (Fig. 12) can be classified into six types: Additive, Digital, Image Attribute Transformation, Weather, Blur, and Physical. These corruptions replicate real-world distortions like noise, compression artifacts, environmental effects, and motion blur, aiding in robustness evaluation and reliability assessment.

14. Dataset Details

The hierarchical distribution of the most frequent two-word phrases across categories is visualized in Fig. 8, which shows the overall distribution and prominence of key phrases within each category.

14.1. Count

The **Count** category is overwhelmingly dominated by variations of "How many", as evident in the sunburst chart in Fig. 8. Other notable phrases include "Number of" and "What is", with significantly lower frequency.

14.2. Order

As seen in the Fig. 8, **Order** related questions are primarily led by "Where is", followed by "Where are" and "What is". The visualization highlights the dominance of location-based questions.

14.3. Trick

Fig. 8 shows that **Trick** category questions frequently begin with "What is", followed by "What can" and "How many", focusing on definitions, possibilities, and numerical inquiries.

14.4. VCR

In the Visual Commonsense Reasoning **VCR** category, "What is" and "Why is" are the most prominent phrases,

Model	#Params	Vision Model	Language Model	Best Inference Method
LLaVA-v1.6 7B [57]	7.06B	CLIP-ViT-L	Vicuna 13B	Corr + TDN
InstructBLIP 7B [19]	7.91B	EVA-G	Vicuna 7B	Corr + TDN
Janus-Pro 7B [13]	7.42B	SigLIP-400M	Deepseek 7B	Corr + VDN + TDN
Idefics2 8B [51]	8.4B	SigLIP-400M	Mistral 7B	Corr + VDN + TDN
Gemini-2.0 Flash [29]	*Closed-Source*			Corr + TDN

Table 5. The table lists each Vision-Language Model (VLM) with its parameter count, underlying vision encoder, language backbone, and the denoising strategy (visual, textual or both) that achieved the best performance under corruption.

as clearly illustrated in Fig. 8. Other notable phrases include "What will" and "Where is", reflecting an emphasis on explanations and predictions.

15. Additional Analysis

Textual Perturbations under Inference Categories. Fig. 13 shows the performance of models across different textual perturbations for each inference category. We observe that Gemini-2.0-Flash exhibits superior robustness to textual perturbations across various inference categories, making it a strong candidate for vision-language tasks in noisy environments. Denoising strategies, such as TDN and VDN, show promise but require further optimization to benefit all models equally. The significant performance drops observed for perturbations like *Homoglyph Substitution*, *Keyboard Based Character Substitution*, and *Synonym Replacement*.

Visual Perturbations under Inference Categories. Fig. 14 shows the performance of models across different visual corruptions for each inference category. We observe that Gemini-2.0 Flash outperforms other models except in the *Rain* noise. The inclusion of a visual denoiser improves Gemini-2.0 Flash's accuracy when dealing with *Rain* noise. In contrast, when the visual denoiser is applied to the *Elastic* noise, Gemini-2.0 Flash exhibits the worst performance, experiencing the most significant accuracy drop compared to other models across all scenarios. Similar observations can be seen for the remaining models, indicating *Elastic* noise being more challenging, even after using denoisers.

Textual Perturbations under Task Categories. Fig. 15 shows model accuracy under textual perturbations across DARE question categories. We observe LLaVA[57] underperforming, compared to other VLMs in most categories. The *Count* category shows the largest fluctuations in model performance across textual corruptions, while the **VCR** category remains mostly stable for all models. Across perturbation types, word- and character-level corruptions, such as *Homoglyph Substitution*, *OCR Character Substitution*, and *Synonym Replacement*, cause the most severe performance drops.

Visual Perturbations under Task Categories. Fig. 16 shows category-wise robustness under visual corruptions. Similar to textual noise, LLaVA is consistently the weakest model, again sharing the lowest performance with InstructBLIP in the **Count** category. Unlike textual perturbations, accuracy trends are smoother across categories, though Gemini suffers larger drops than other models. Among visual corruptions, **Elastic**, **Rain**, **Shot**, and **Zoom Blur** are the most detrimental, consistently degrading performance across categories.

Model-wise Error Analysis. Fig. 10 illustrates variations of VLM performance under varying input conditions. Some models show natural robustness to visual and textual noise, which can be attributed to the model's inherent robustness. Others work well with the denoisers, contributing to the better system robustness.

16. Unanswerability of Questions

Severe visual or textual corruption can make certain questions inherently unanswerable, a problem also noted by VRB [40]. In such cases, the performance drop cannot be attributed solely to the system's shortcomings in robustness, but rather to the inherent unanswerability of the input. We account for this by establishing human baselines in Sec. 6 as a reference for task answerability.

Human annotators achieve an average accuracy of roughly 80% across the dataset, with scores being consistent across categories, indicating that even humans incur a roughly 20% accuracy drop under multimodal corruptions. We can expect models to face similar performance degradation when subjected to such corruption effects. However, as the overall human performance remains high, we can conclude that the dataset is still largely answerable despite the severity of the multimodal corruptions.

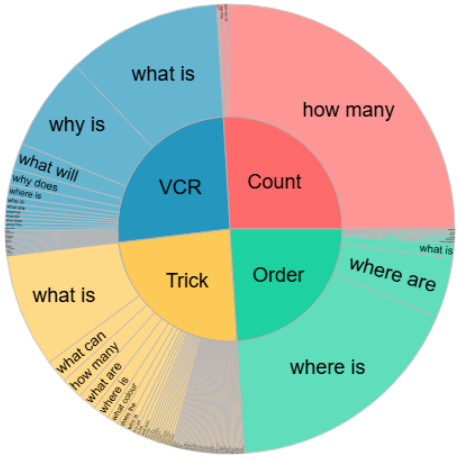


Figure 8. A breakdown of the DARE dataset in single-correct answer evaluation scenarios.

Zero-Shot Prompt for VLM Inference

The following are multiple-choice questions about `<QUESTION TYPE>`. You should directly answer the question by choosing the correct option, given the image and the question. Give only the letter indicating the correct answer (e.g., "A").

Question: `<QUESTION>`

Options:
A. `<ANSWER A>`
B. `<ANSWER B>`
C. `<ANSWER C>`
D. `<ANSWER D>`

Answer:

Figure 9. VLM inference prompt template from DARE [74].

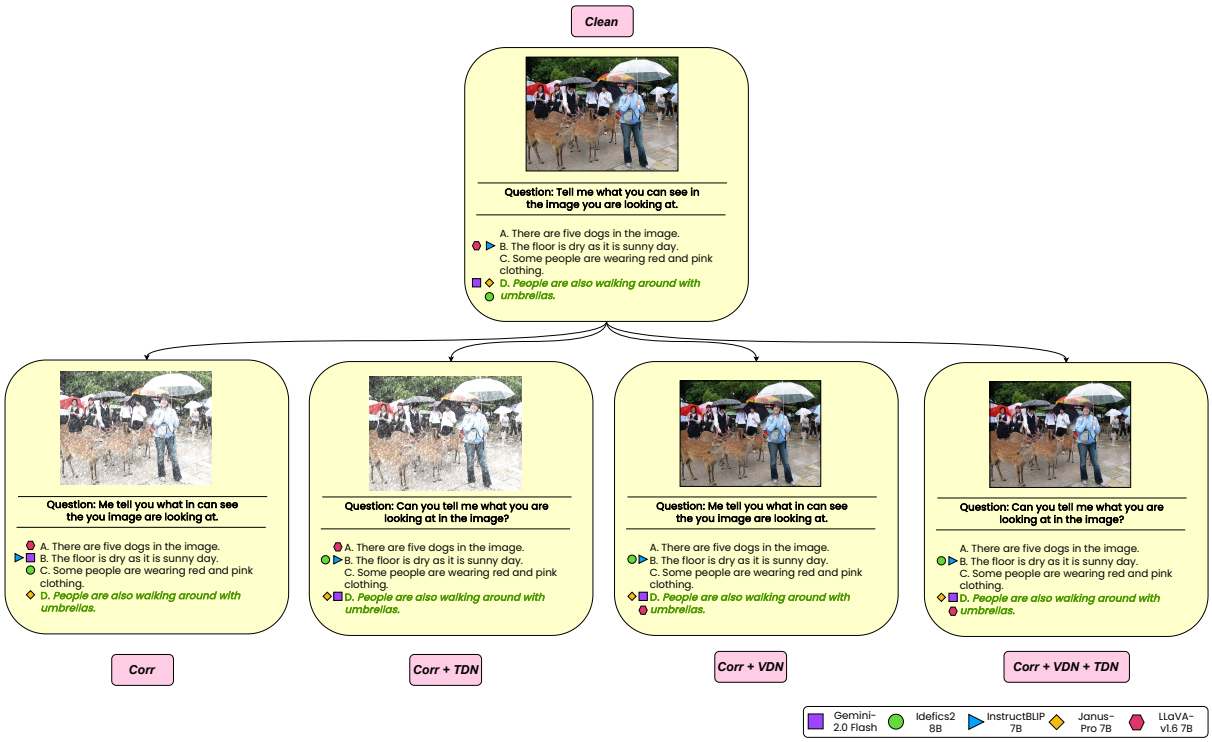
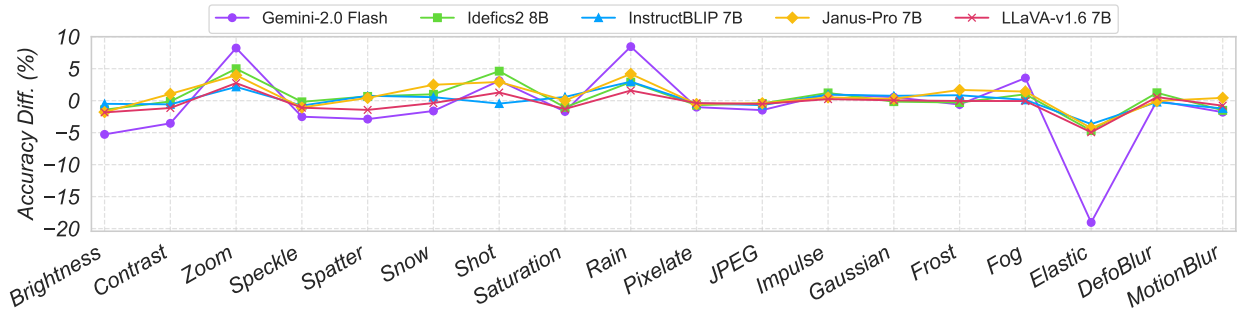
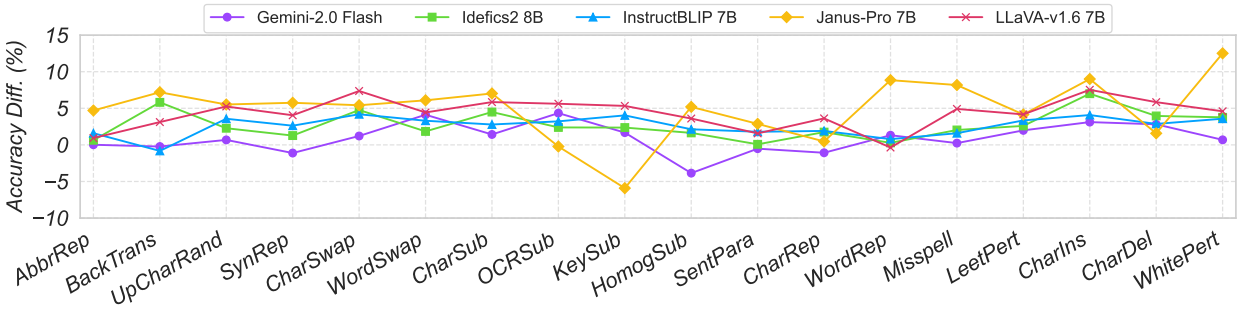


Figure 10. VLM responses under input variations: Clean (no noise), Corr (both modalities corrupted), Corr + TDN (image corrupted, text denoised), Corr + VDN (text corrupted, image denoised), and Corr + VDN + TDN (both denoised).



(a) Average accuracy improvement of VLMs after applying only visual denoiser (VDN) across 18 visual corruption types, measured as (Corr + VDN) − Corr.



(b) Average accuracy improvement of VLMs after applying only textual denoiser (TDN) across 18 textual corruption types, measured as (Corr + TDN) − Corr.

Figure 11. Average accuracy improvement of VLMs after applying denoisers. Top: visual denoiser. Bottom: textual denoiser.

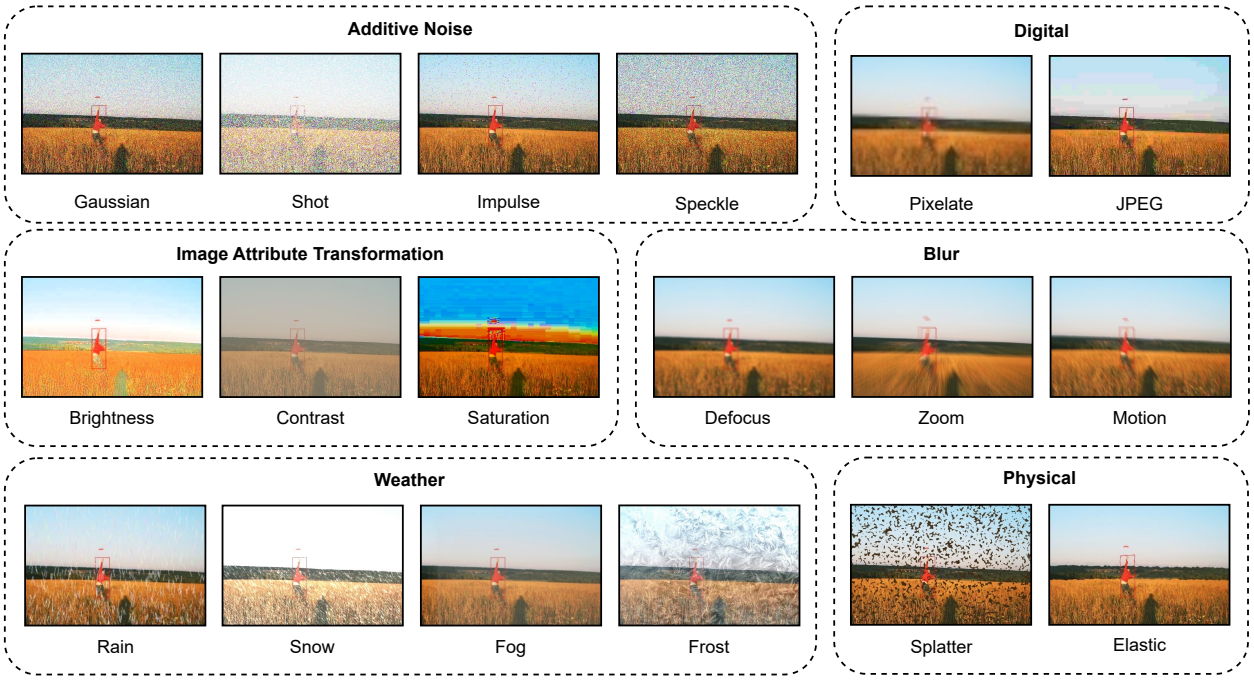
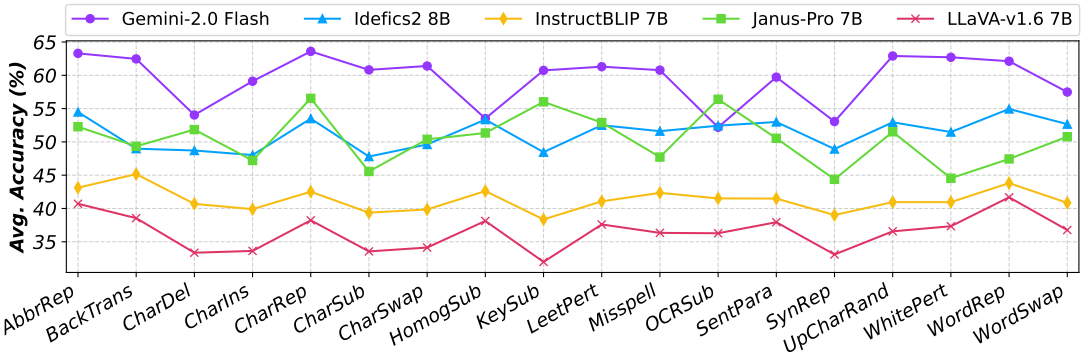


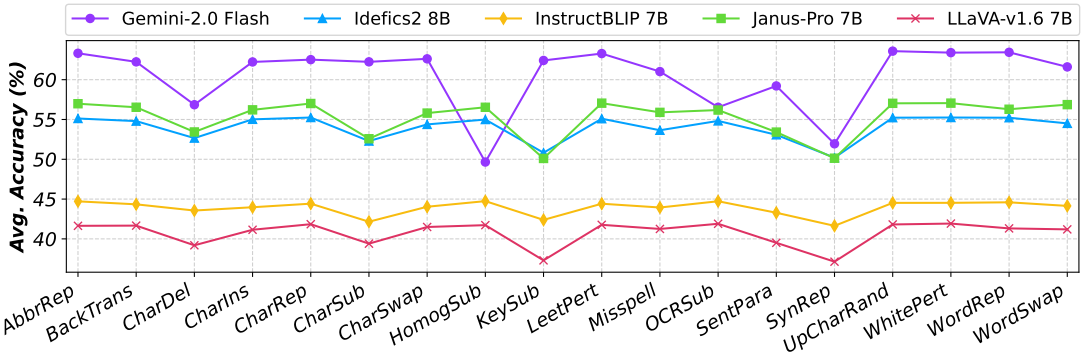
Figure 12. Types of Visual Corruptions distributed into corresponding classes

Noise Category	Perturbation Type	Noisy Question	Denoised Question
Word	Abbreviation Replacement	where r the 2 blue lego figures in the pic?	where are the two blue lego figures in the picture?
	Synonym Replacement	where are the two azure lego statues in the photo?	where are the two blue lego statues in the picture?
	Random Word Swapping	where the two blue lego figures are in the picture?	where are the two blue lego figures in the picture?
	Word Repetition	where are the two two blue lego lego figures in in the picture?	where are the two blue lego figures in the picture?
	Misspelling Words	whrre ar the too blue lego figuress in the picturre?	where are the two blue lego figures in the picture?
Sentence	Back Translation	where are the two azure lego figurines in the image?	where are the two blue lego figurines in the image?
	Sentence Paraphrasing	Can you tell me the location of the two blue lego figures?	Can you tell me where the two blue lego figures are?
Character	Uppercase Char Randomly	WheRe aRe tHe TwO bLuE LeGo fiGuReS iN tHe picTure?	where are the two blue lego figures in the picture?
	Random Character Swapping	wheer are the tow blue lego figures in the pictuer?	
	Homoglyph Substitution	where are two blue lego figures in the picture?	
	Random Character Substitution	whera are the two blue legi figures on the pictkre?	
	Substitute Char by OCR	where are the two b1ue lego fi9ures in the picture?	
	Keyboard-based Char Substitution	whwre ate the ywo blie legi fihures in the oicture?	
	Random Character Repetition	wherree arre the twoo blueee leggoo figureees in the picturee?	
	Leetspeak with Perturbation	wh3r3 4r3 th3 tw0 blu3 l3g0 figur3s 1n th3 p1ctur3?	
	Random Character Insertion	whexre are the twqo blue lego figyures in the piczture?	
	Random Character Deletion	wher ar the two blue lego figure in the pictur?	
	Whitespace Perturbation	wher e are t he tw o blu e lego figures in th e picture?	

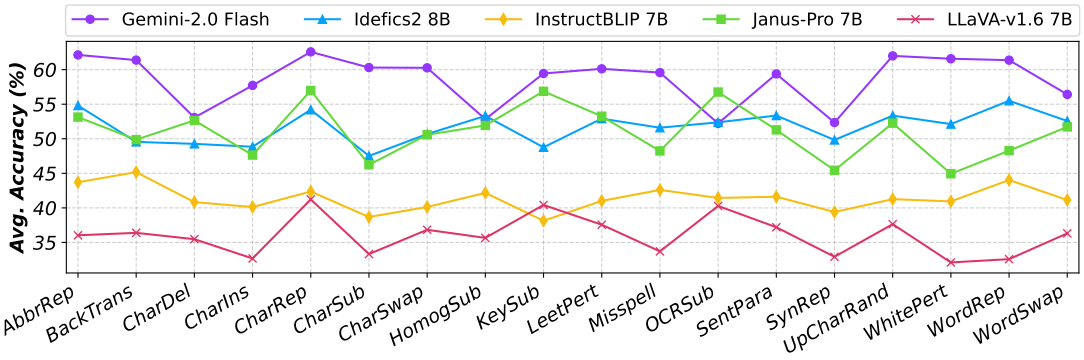
Table 8. Examples of corrupted questions and their denoised counterparts, categorized by corruption type.



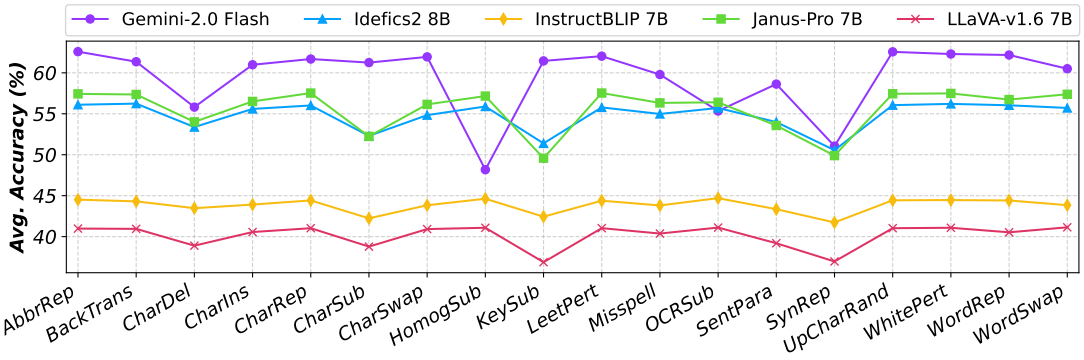
(a) **Corr**



(b) **Corr + TDN**

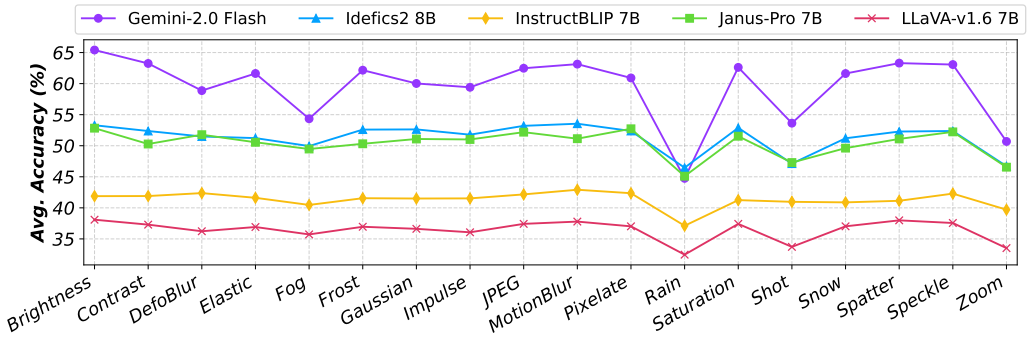


(c) **Corr + VDN**

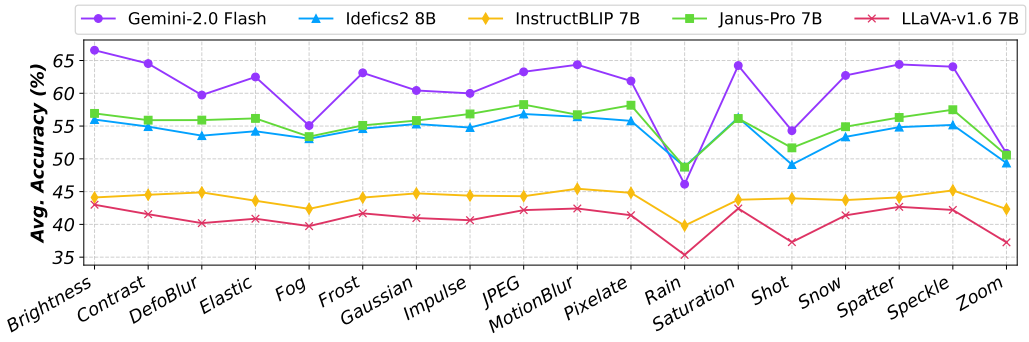


(d) **Corr + VDN + TDN**

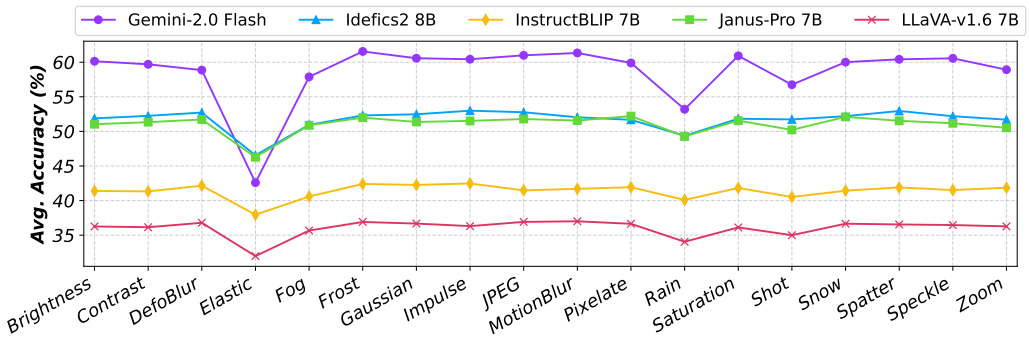
Figure 13. Average accuracy of models across different textual perturbations under four inference categories.



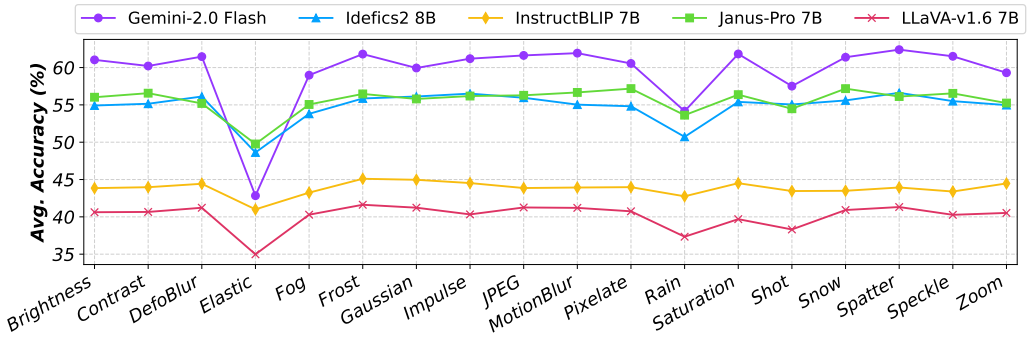
(a) **Corr**



(b) **Corr + TDN**

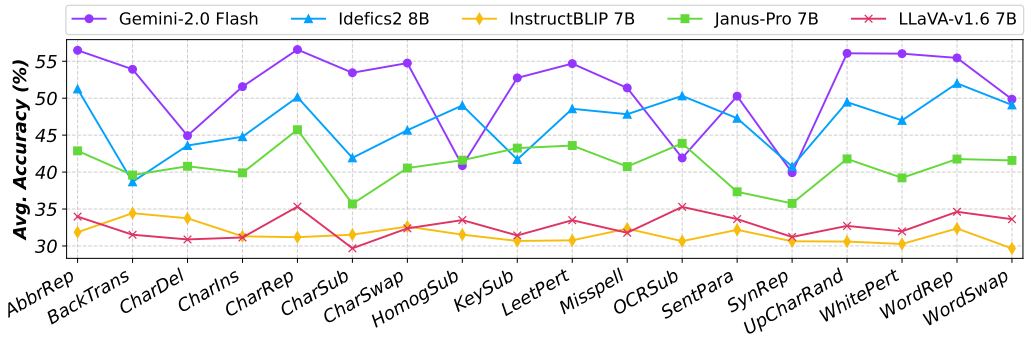


(c) **Corr + VDN**

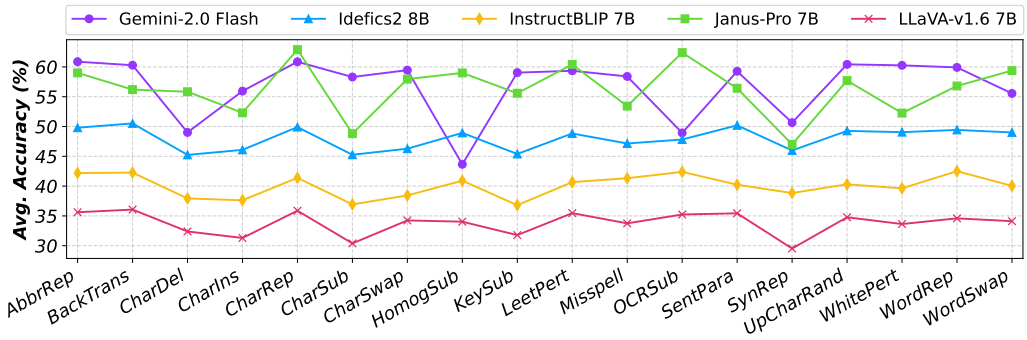


(d) **Corr + VDN + TDN**

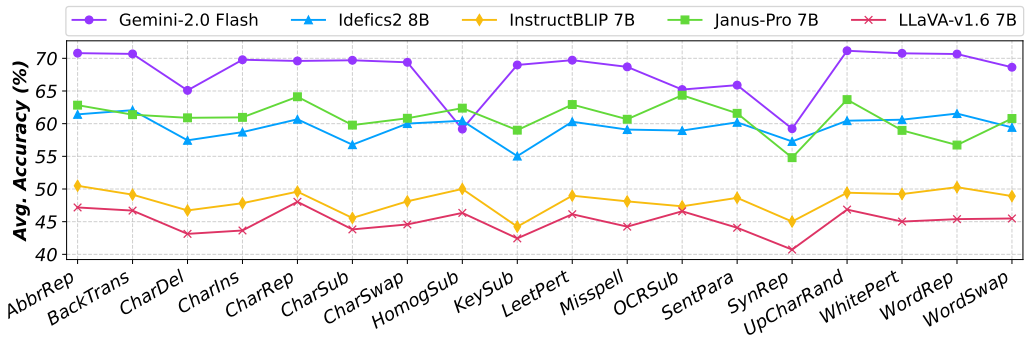
Figure 14. Average accuracy of models across different visual corruptions under four inference categories.



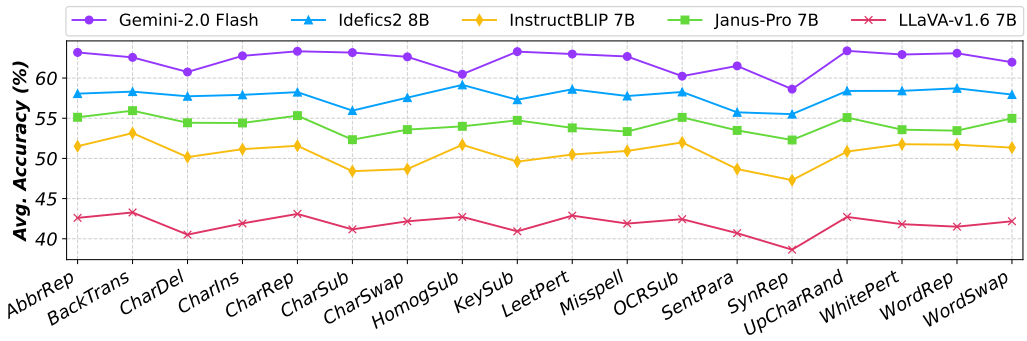
(a) **Count**



(b) **Order**

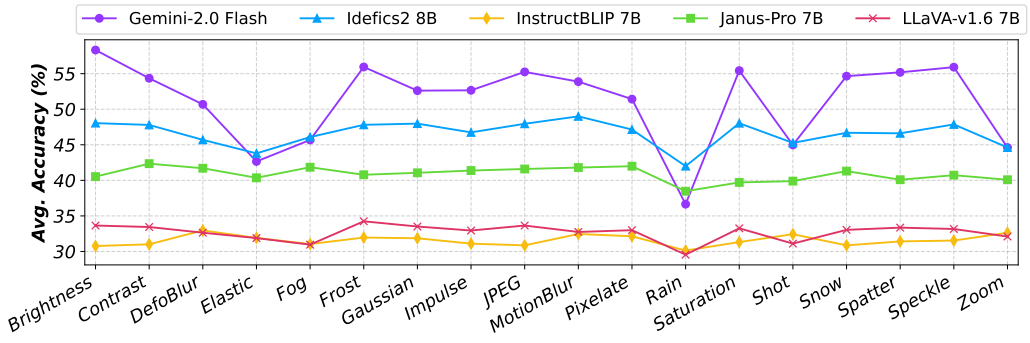


(c) **Trick**

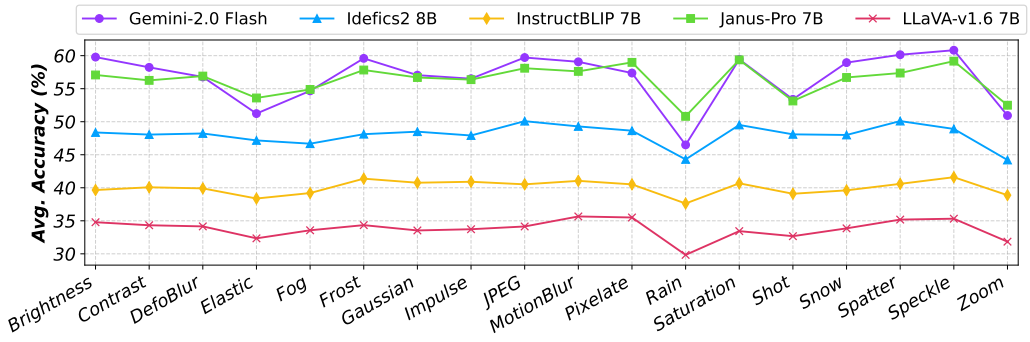


(d) **VCR**

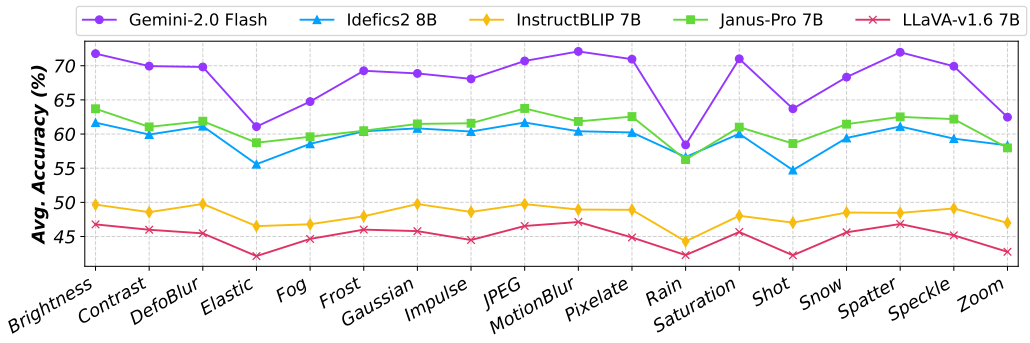
Figure 15. Average accuracy of models across different textual perturbations under four DARE question categories.



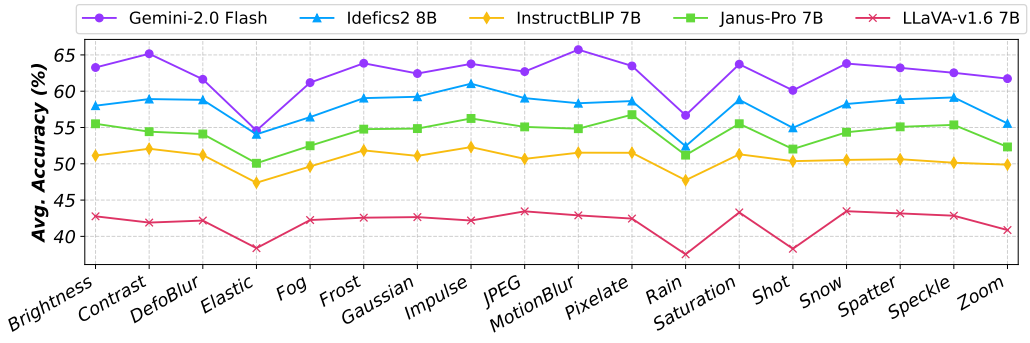
(a) **Count**



(b) **Order**

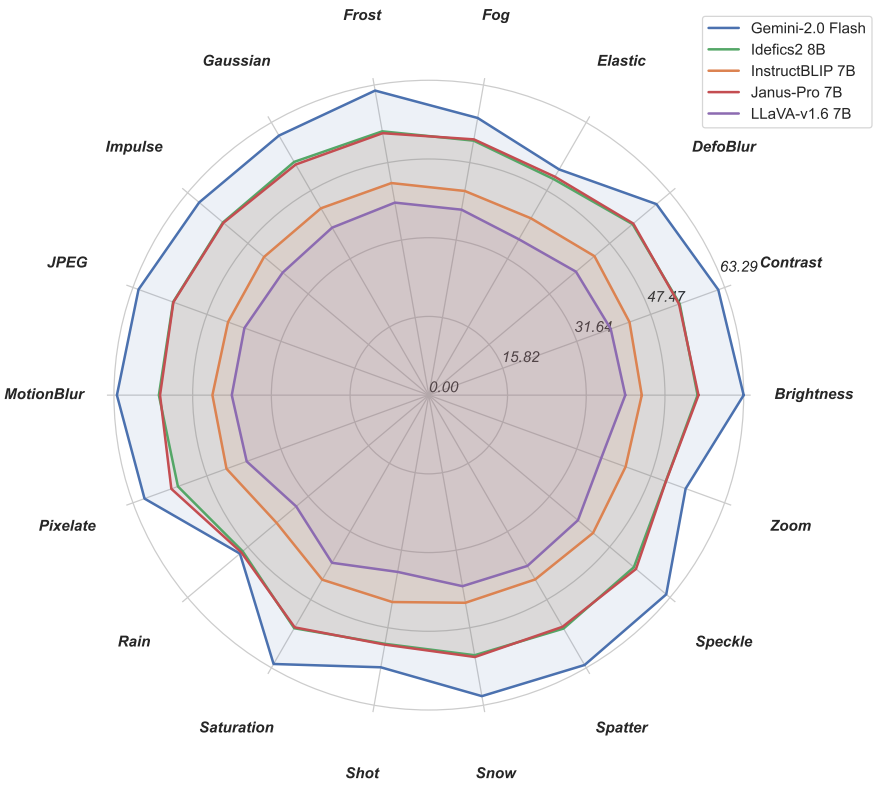


(c) **Trick**

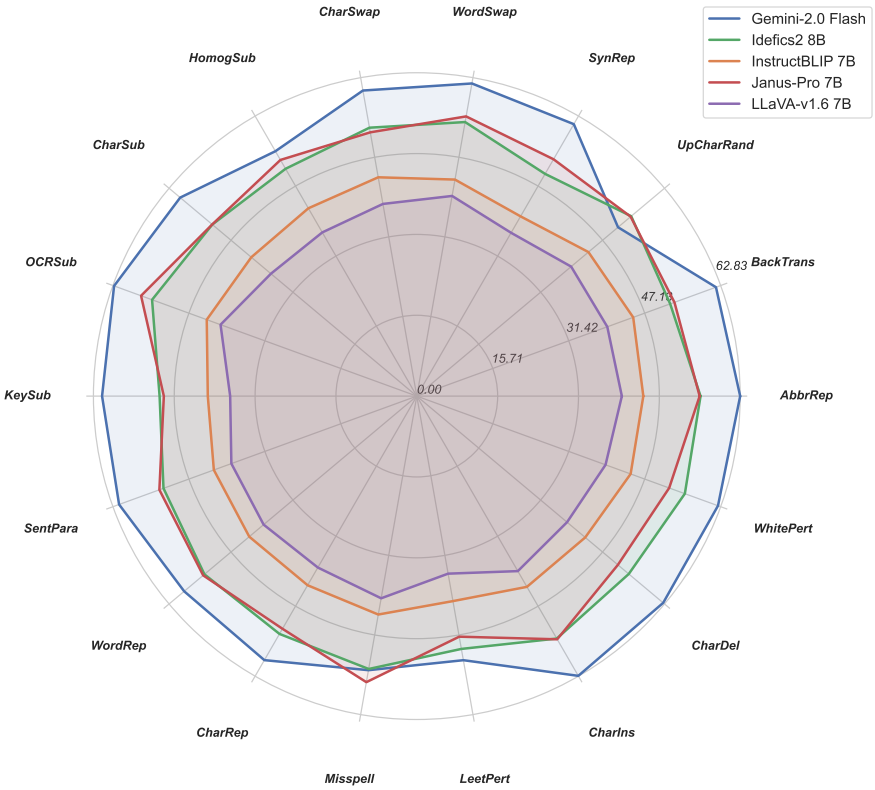


(d) **VCR**

Figure 16. Average accuracy of models across different visual corruptions under four DARE question categories.



(a) Image Corruption vs Model's Average Accuracy



(b) Textual Perturbation vs Model's Average Accuracy

Figure 17. Radar chart showing the Average accuracy of models across different perturbations scenario

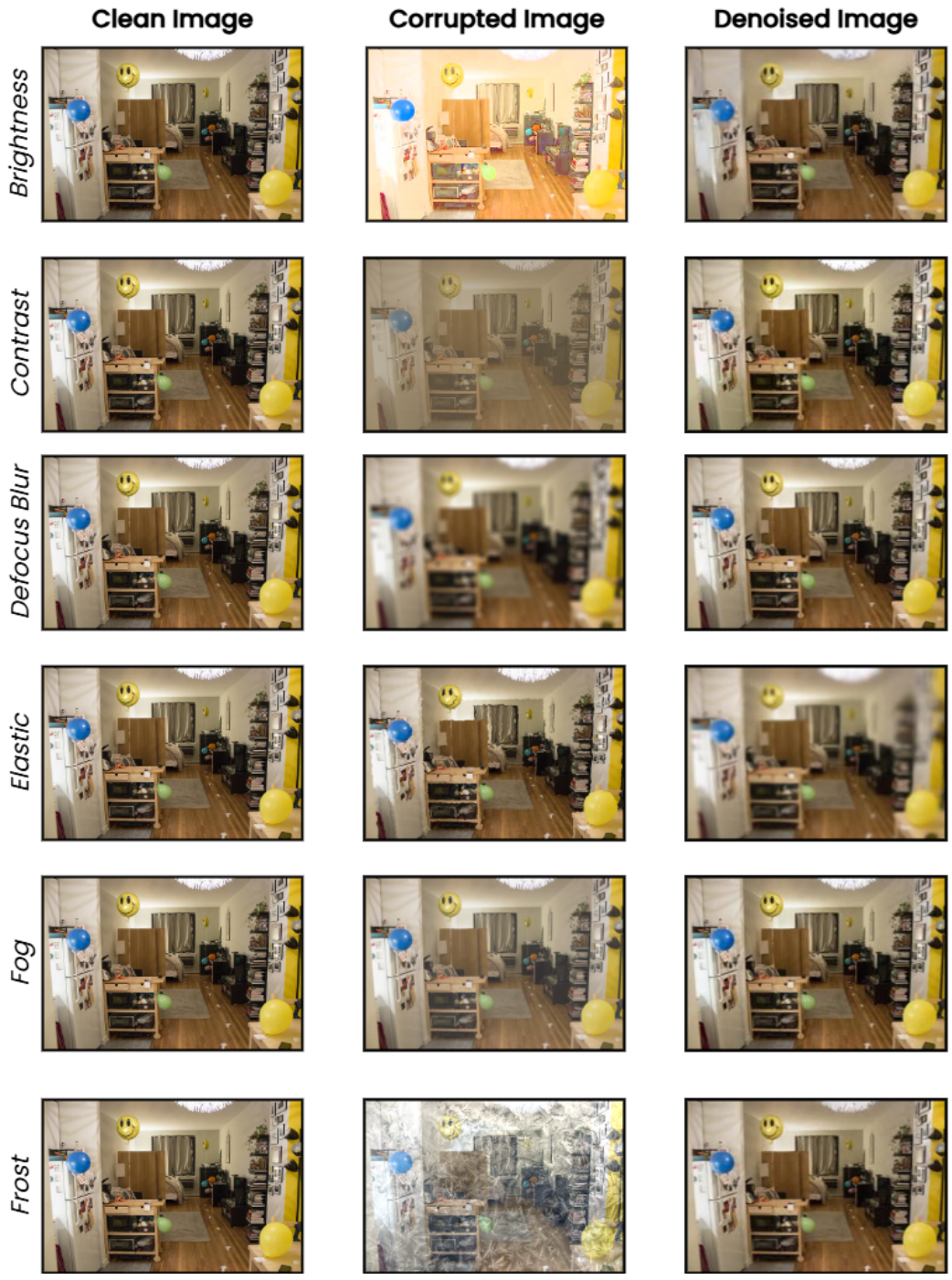


Figure 18. Clean, Corrupted, and Denoised images for the first set of 6 corruption types

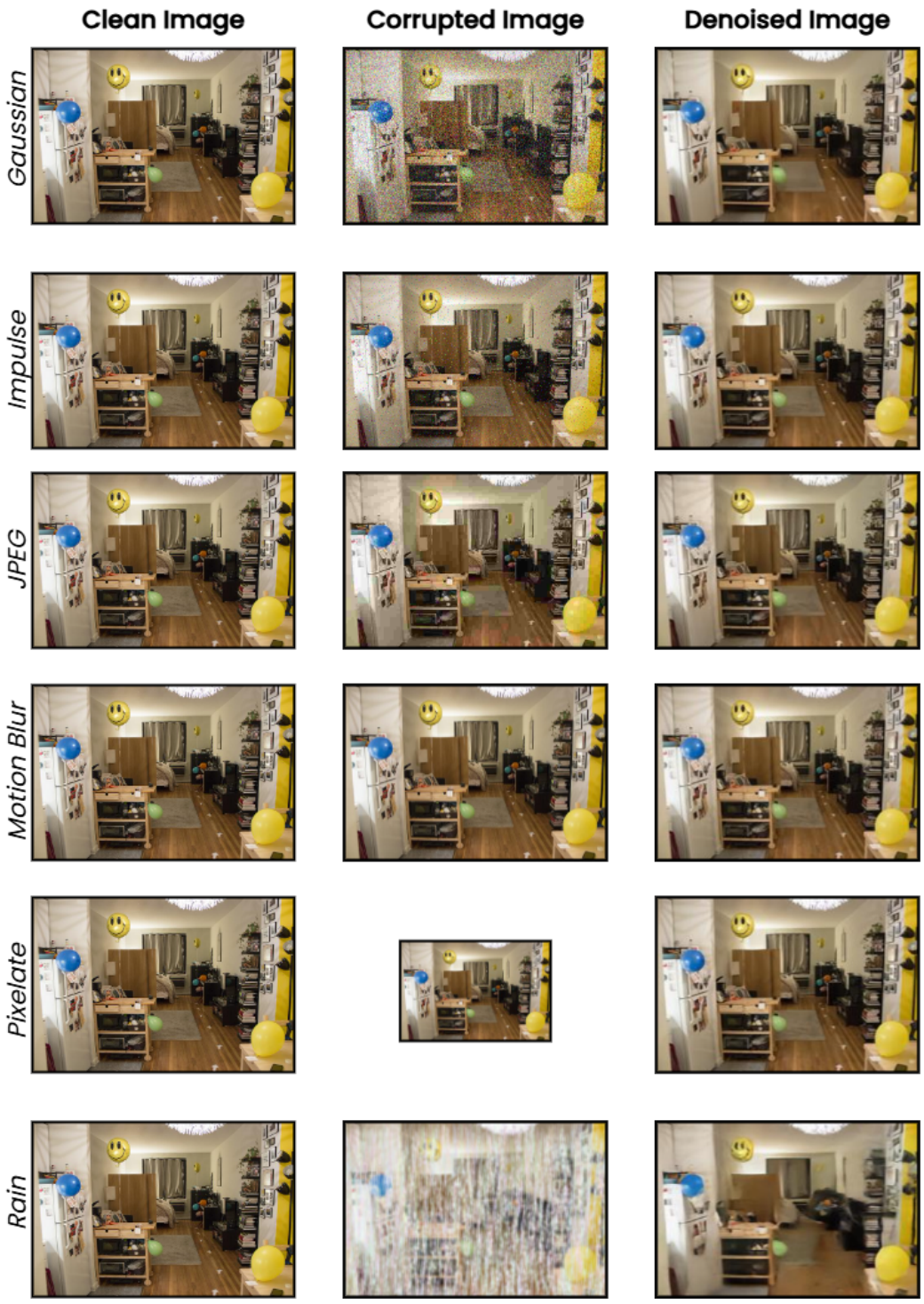


Figure 19. Clean, Corrupted, and Denoised images for the second set of 6 corruption types

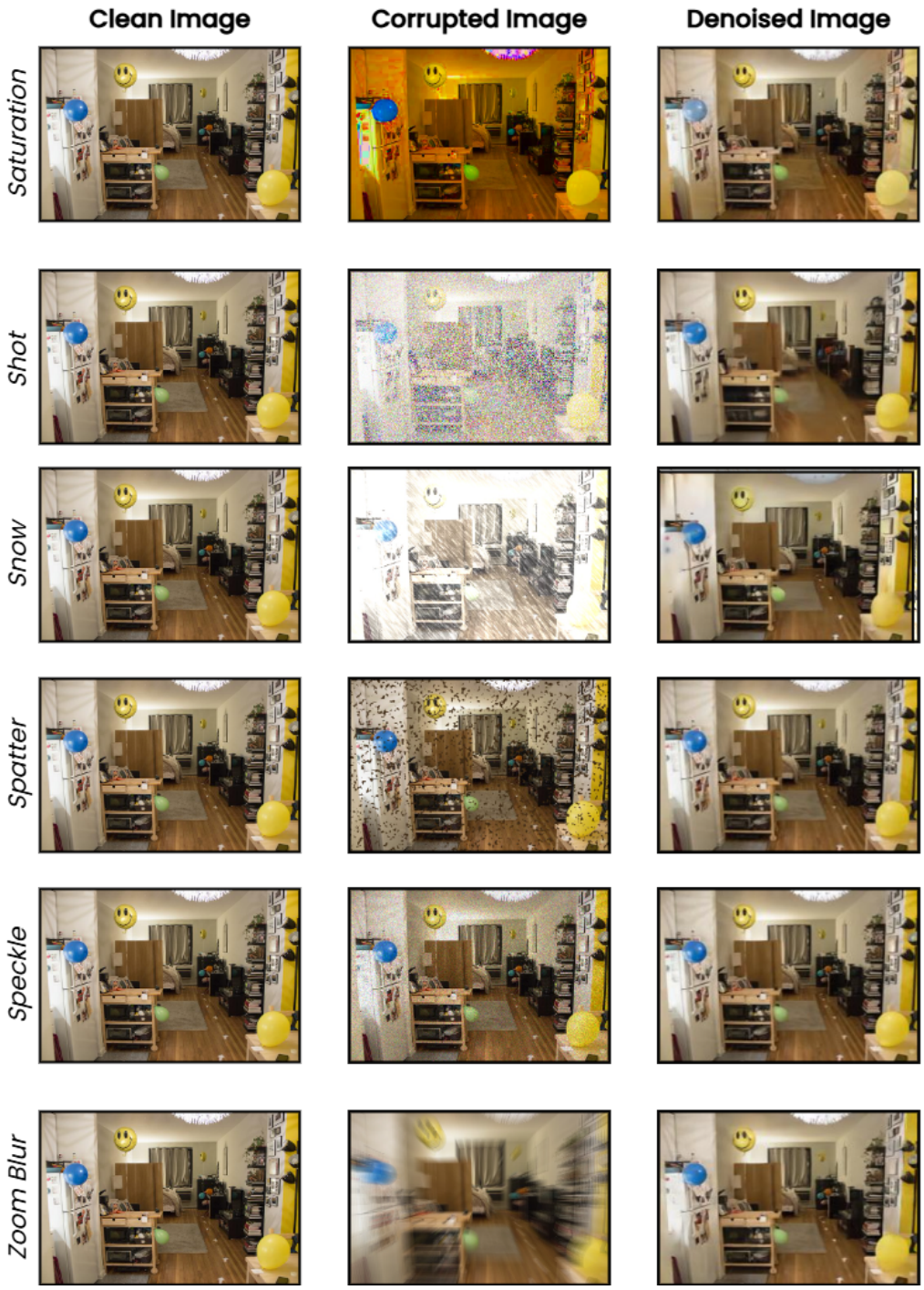


Figure 20. Clean, Corrupted, and Denoised images for the third set of 6 corruption types

Model	Noise	Full-Reference Metrics			No-Reference Metrics	
		PSNR ↑	SSIM ↑	LPIPS ↓	CLIP ↑	BRISQUE ↓
DRUNet	Brightness	26.41	0.880	0.061	0.834	19.49
	Contrast	26.11	0.890	0.063	0.829	19.08
	Defocus-blur	30.49	0.900	0.055	0.796	21.01
	Elastic	18.83	0.510	0.425	0.422	48.04
	Fog	24.89	0.820	0.137	0.656	26.89
	Frost	28.66	0.900	0.055	0.823	19.43
	Gaussian	30.13	0.880	0.067	0.807	22.06
	Impulse	30.15	0.890	0.064	0.811	21.82
	JPEG-compression	30.50	0.900	0.056	0.830	20.02
	Pixelate	29.74	0.890	0.071	0.785	23.59
	Rain	24.88	0.750	0.158	0.584	27.90
	Saturation	27.91	0.890	0.076	0.821	19.43
	Shot	27.87	0.820	0.121	0.724	24.28
	Snow	26.95	0.860	0.085	0.813	21.21
	Spatter	30.52	0.910	0.052	0.829	19.63
	Speckle	30.35	0.900	0.066	0.829	20.73
	Zoom-blur	27.14	0.840	0.097	0.763	20.51
	Motion-blur	30.30	0.900	0.055	0.804	20.16
DnCNN	Brightness	22.41	0.830	0.093	0.844	18.44
	Contrast	19.40	0.750	0.191	0.794	18.43
	Defocus-blur	28.64	0.860	0.096	0.668	25.17
	Elastic	18.36	0.470	0.381	0.315	41.63
	Fog	17.29	0.590	0.420	0.434	49.96
	Frost	21.67	0.720	0.193	0.703	18.30
	Gaussian	29.41	0.860	0.096	0.687	20.90
	Impulse	29.40	0.860	0.098	0.672	21.92
	JPEG-compression	30.26	0.900	0.064	0.798	19.77
	Pixelate	29.48	0.880	0.078	0.749	23.75
	Rain	22.77	0.650	0.279	0.458	29.26
	Saturation	24.61	0.860	0.103	0.794	19.70
	Shot	26.30	0.760	0.189	0.451	23.71
	Snow	24.09	0.780	0.143	0.709	20.86
	Spatter	29.89	0.890	0.064	0.776	19.99
	Speckle	29.78	0.880	0.078	0.761	19.99
	Zoom-blur	21.36	0.660	0.227	0.506	21.60
	Motion-blur	27.90	0.850	0.100	0.661	28.25
BRDNet	Brightness	22.94	0.840	0.085	0.847	17.98
	Contrast	20.34	0.780	0.159	0.789	20.91
	Defocus-blur	28.75	0.860	0.095	0.712	23.72
	Elastic	18.47	0.480	0.395	0.341	46.34
	Fog	20.60	0.710	0.250	0.640	32.18
	Frost	21.40	0.730	0.185	0.677	18.52
	Gaussian	29.42	0.860	0.091	0.704	19.99
	Impulse	29.42	0.860	0.094	0.680	19.71
	JPEG-compression	30.24	0.900	0.066	0.792	19.99
	Pixelate	29.46	0.880	0.078	0.757	22.99
	Rain	22.87	0.660	0.277	0.419	32.32
	Saturation	24.55	0.860	0.101	0.811	19.76
	Shot	26.40	0.770	0.181	0.461	24.09
	Snow	22.57	0.750	0.177	0.640	20.40
	Spatter	29.64	0.890	0.066	0.760	20.40
	Speckle	29.72	0.880	0.079	0.760	19.46
	Zoom-blur	22.35	0.690	0.222	0.524	25.61
	Motion-blur	27.87	0.850	0.099	0.686	27.80

Table 9. Performance of denoisers under different noise types.

Figure 21. Examples of four worst types of Visual Corruptions (Elastic, Rain, Shot, and Fog) and their effects after applying our visual denoiser. The images show the original, corrupted, and denoised versions, highlighting residual artifacts that remain after denoising.