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 VGG in terms of PSNR and SSIM, and VGG outperforming MSE in terms of LPIPS, both indicating better perceptual quality. LPIPS loss surprisingly performs worse than VGG Loss in its own metric, but outperforms the rest in CLIP-IQA and BRISQUE, indicating a great alignment of the visual content.

Model	Full Reference Metrics			No Reference Metrics	
	PSNR ↑	SSIM ↑	LPIPS ↓	CLIP ↑	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 like 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.

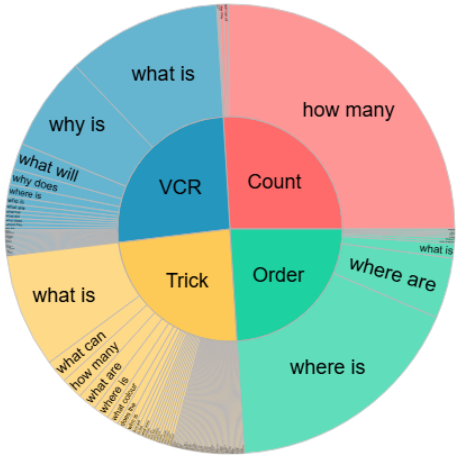


Figure 8. A breakdown of the DARE dataset in single-correct answer evaluation scenarios.

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].

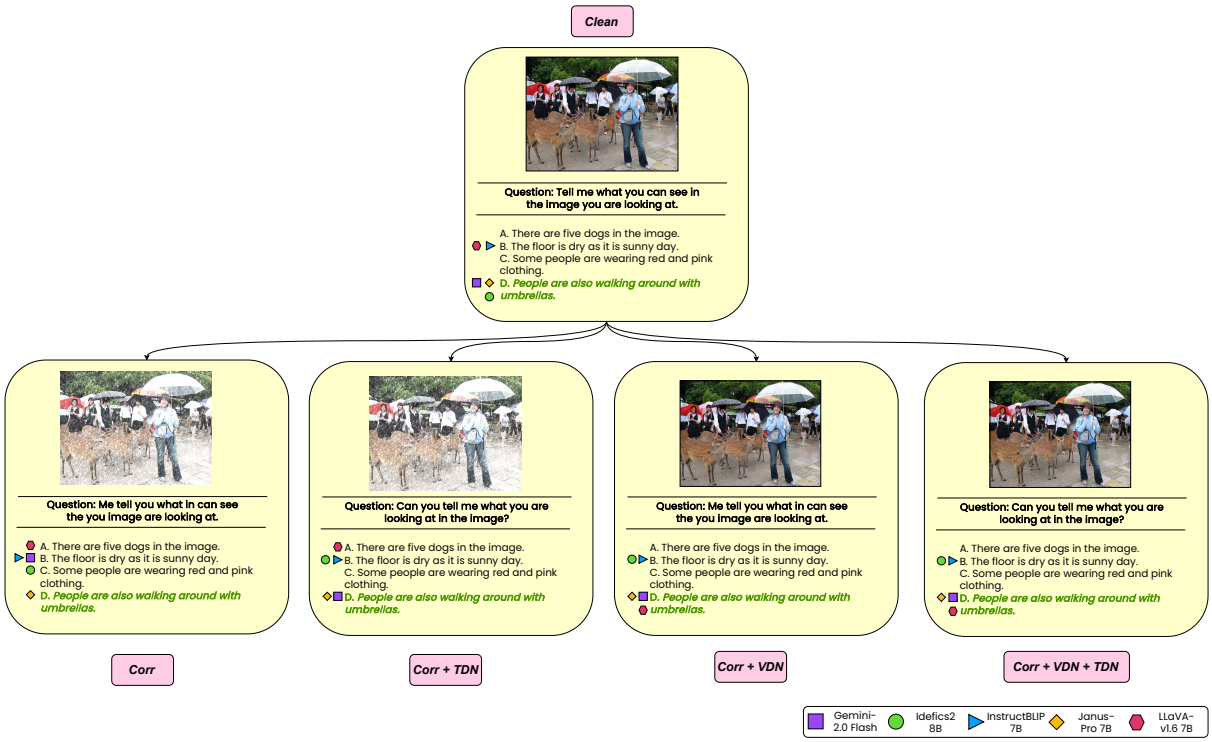
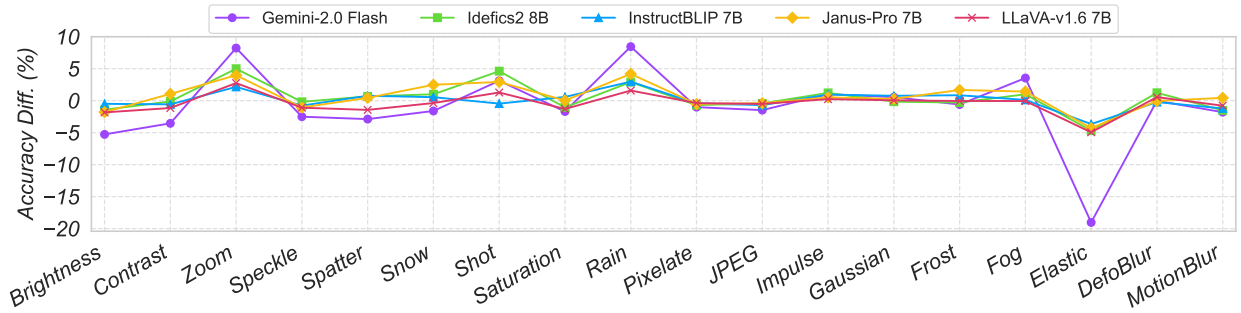
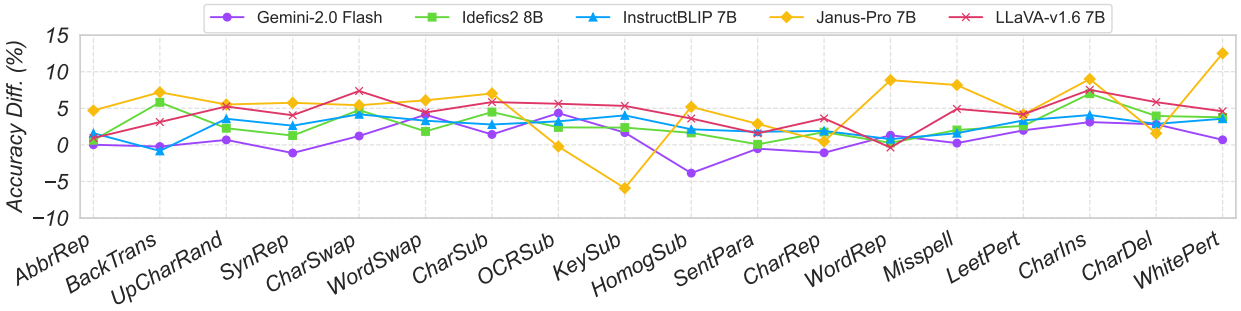


Figure 10. VLM responses under input variations: Clean (no noise), Corr (both modalities corrupted), Corr + TDN (image corrupted, text denoised), Corr + VDN (text corrupted, image denoised), and Corr + VDN + TDN (both denoised).



(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.

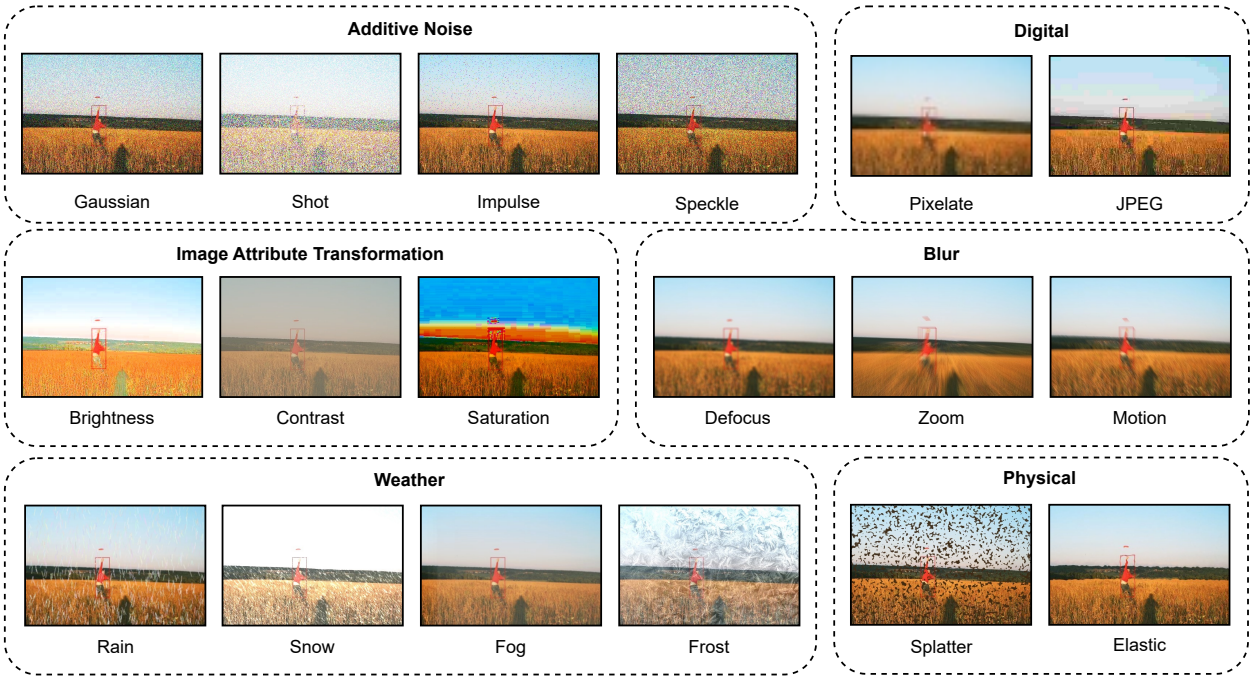
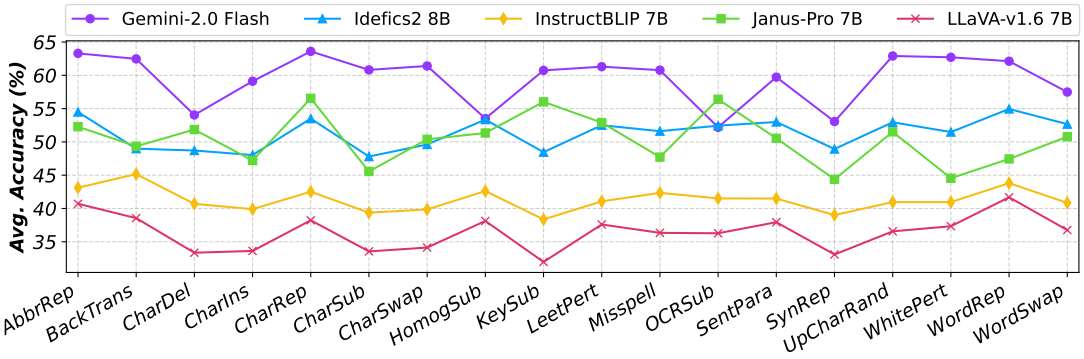


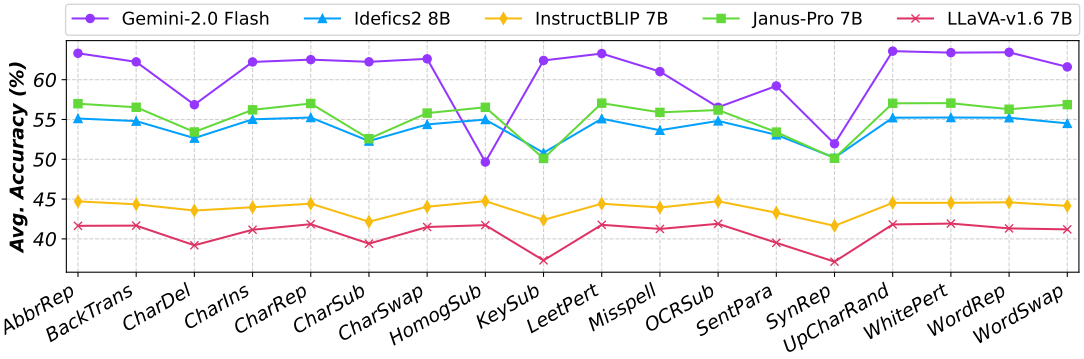
Figure 12. The visual corruptions in our benchmark, categorized into corruption classes.

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?	

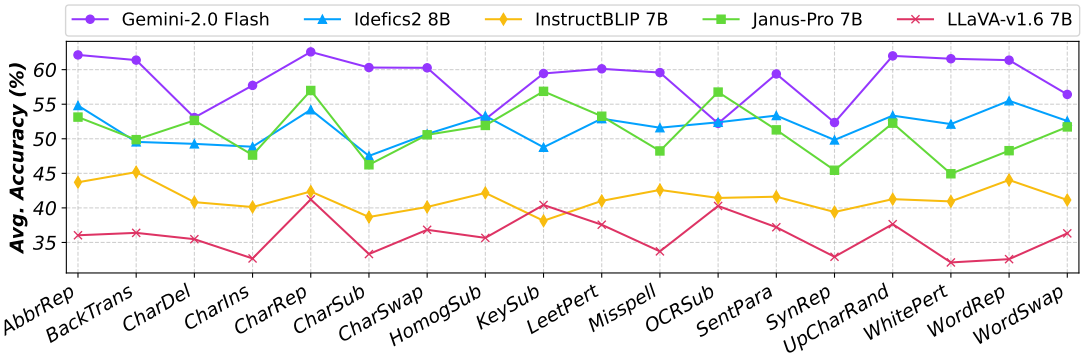
Table 8. Examples of corrupted questions and their denoised counterparts, categorized by corruption types.



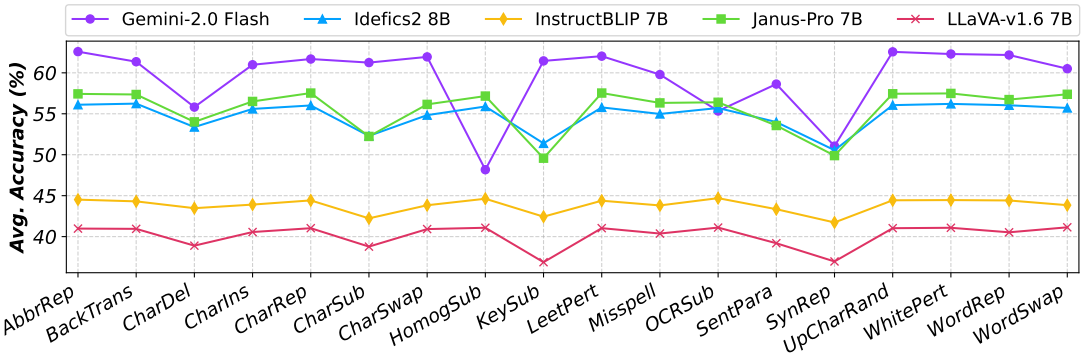
(a) **Corr**



(b) **Corr + TDN**

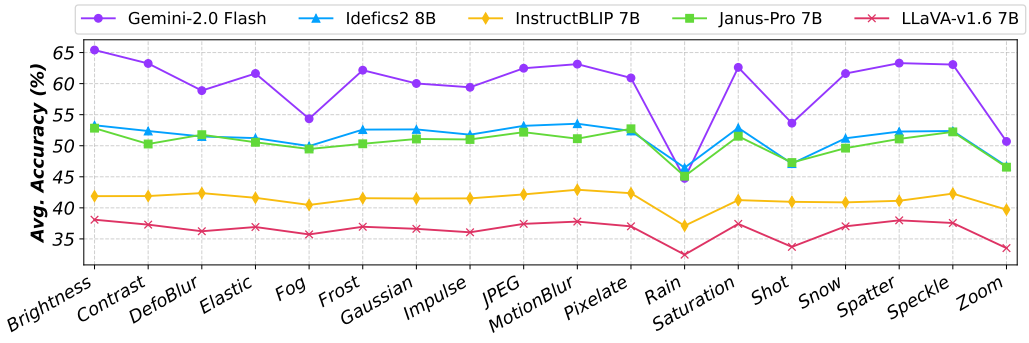


(c) **Corr + VDN**

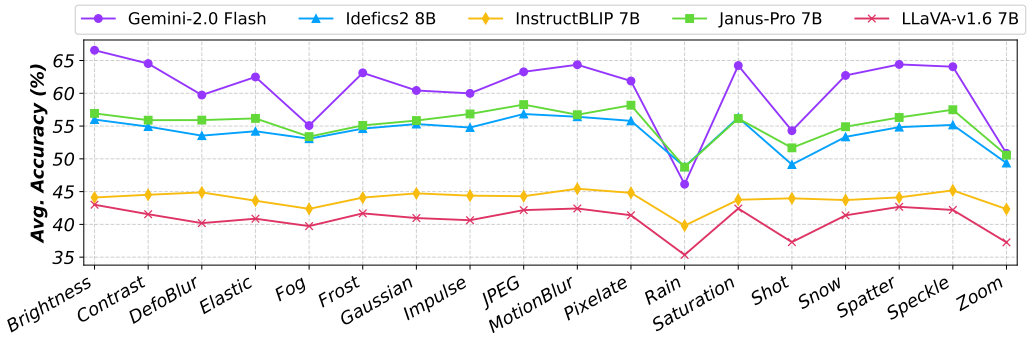


(d) **Corr + VDN + TDN**

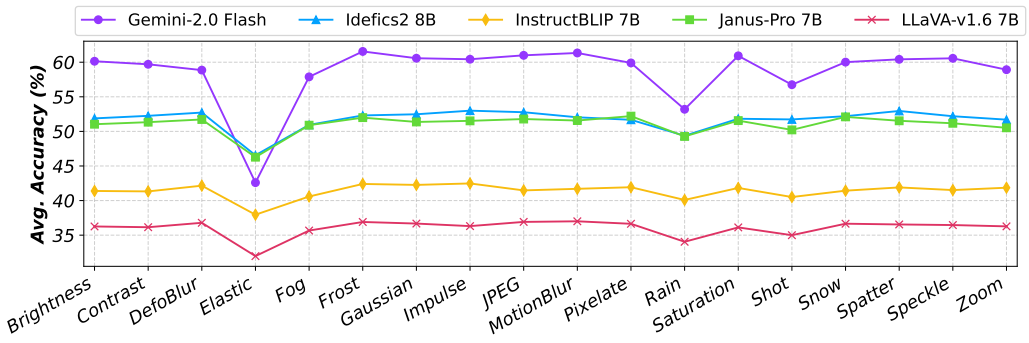
Figure 13. Average accuracy of models across different textual perturbations under four inference categories.



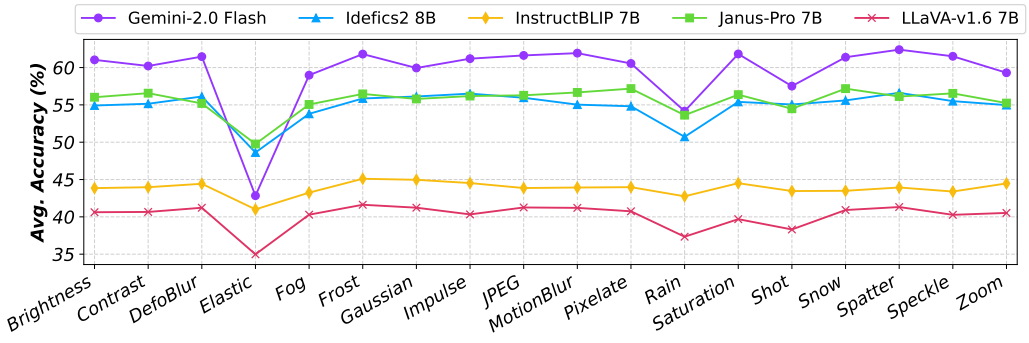
(a) **Corr**



(b) **Corr + TDN**

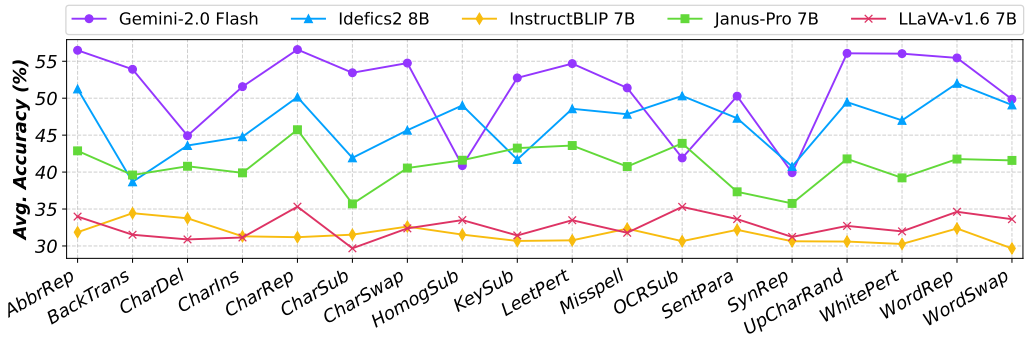


(c) **Corr + VDN**

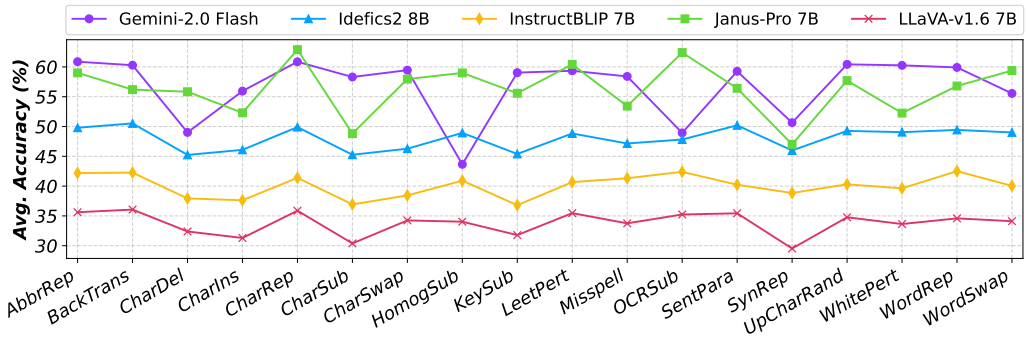


(d) **Corr + VDN + TDN**

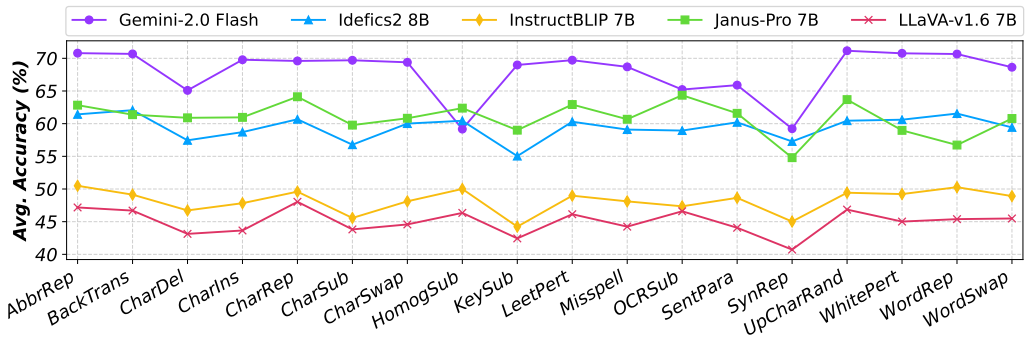
Figure 14. Average accuracy of models across different visual corruptions under four inference categories.



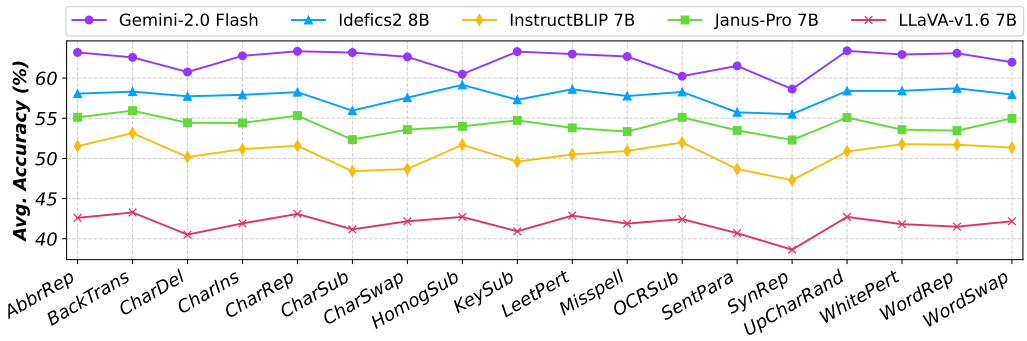
(a) **Count**



(b) **Order**

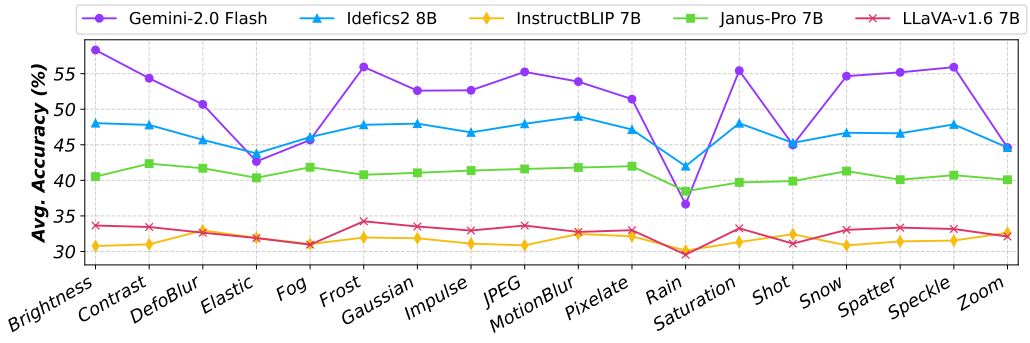


(c) **Trick**

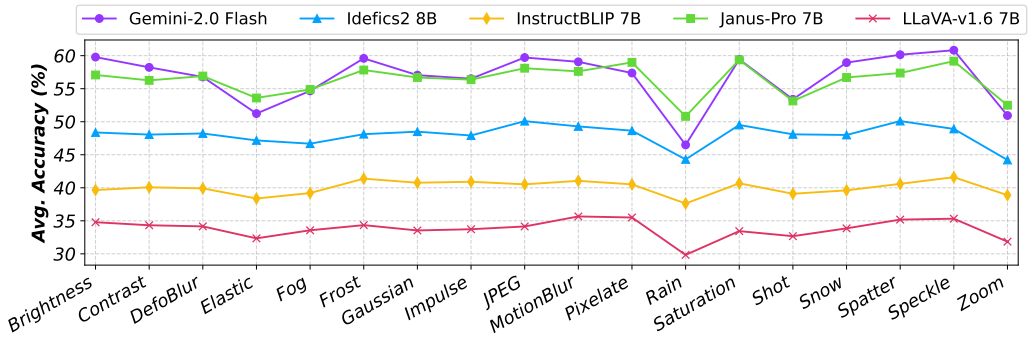


(d) **VCR**

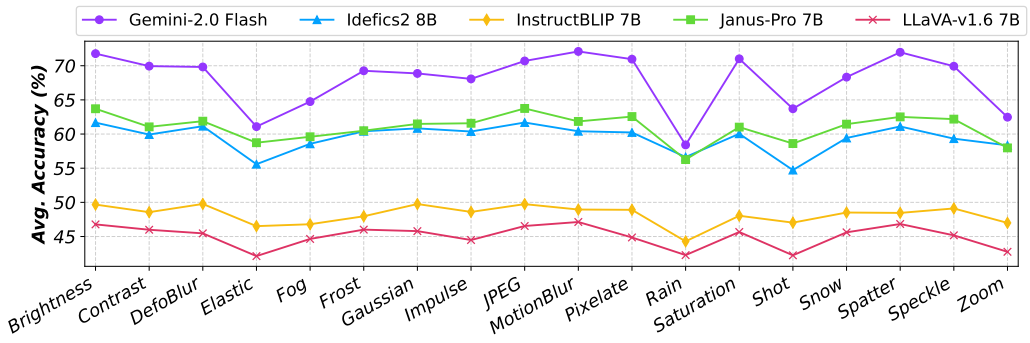
Figure 15. Average accuracy of models across different textual perturbations under four DARE question categories.



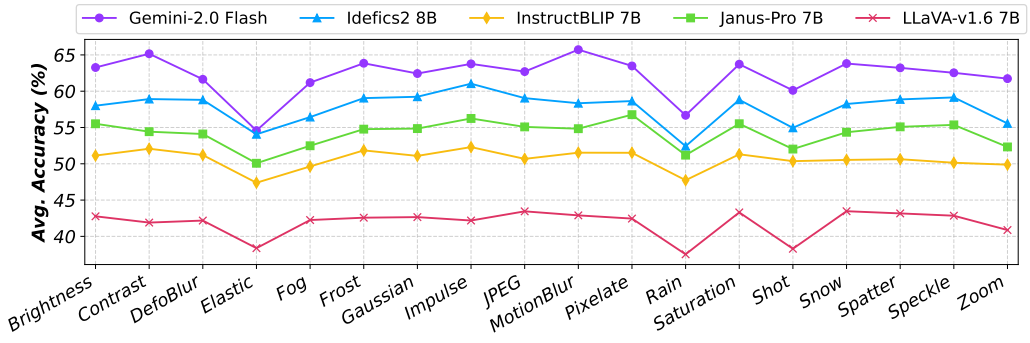
(a) **Count**



(b) **Order**

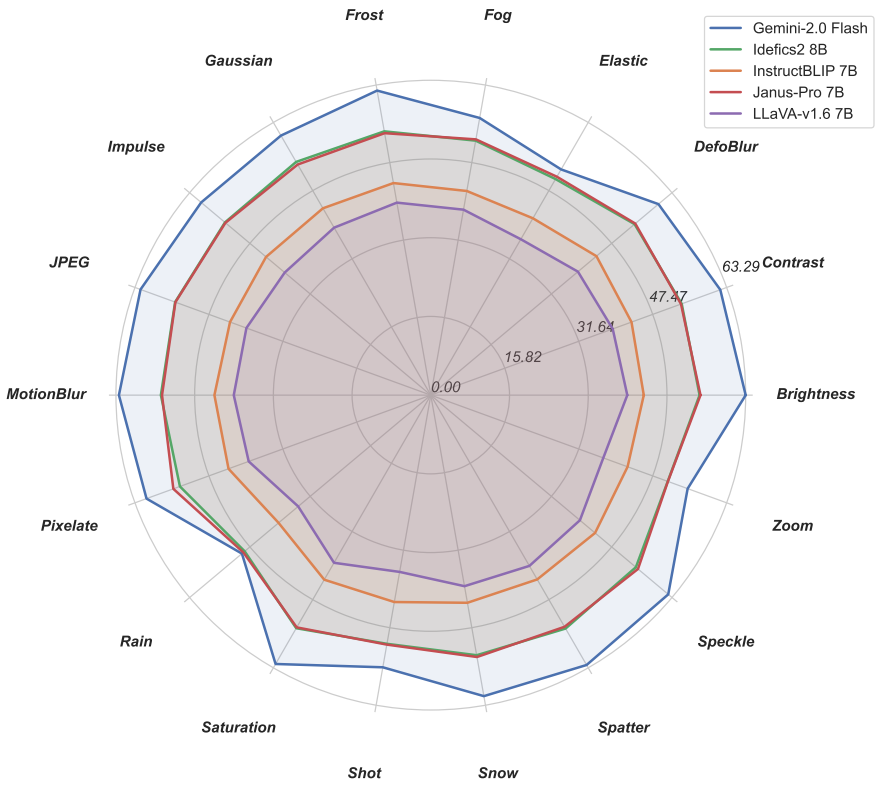


(c) **Trick**

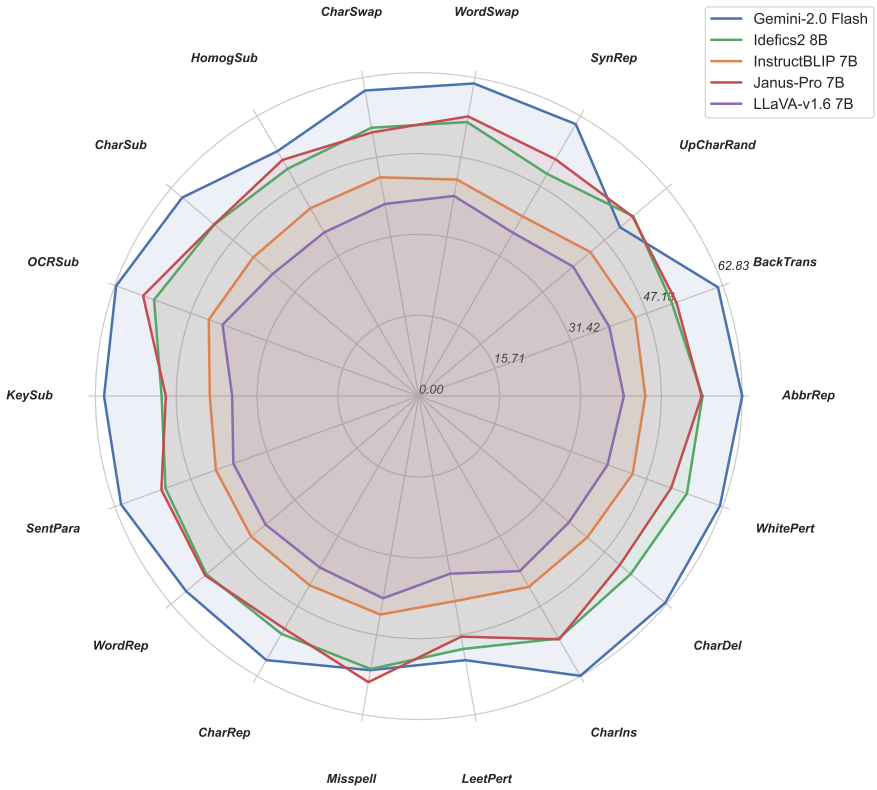


(d) **VCR**

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.

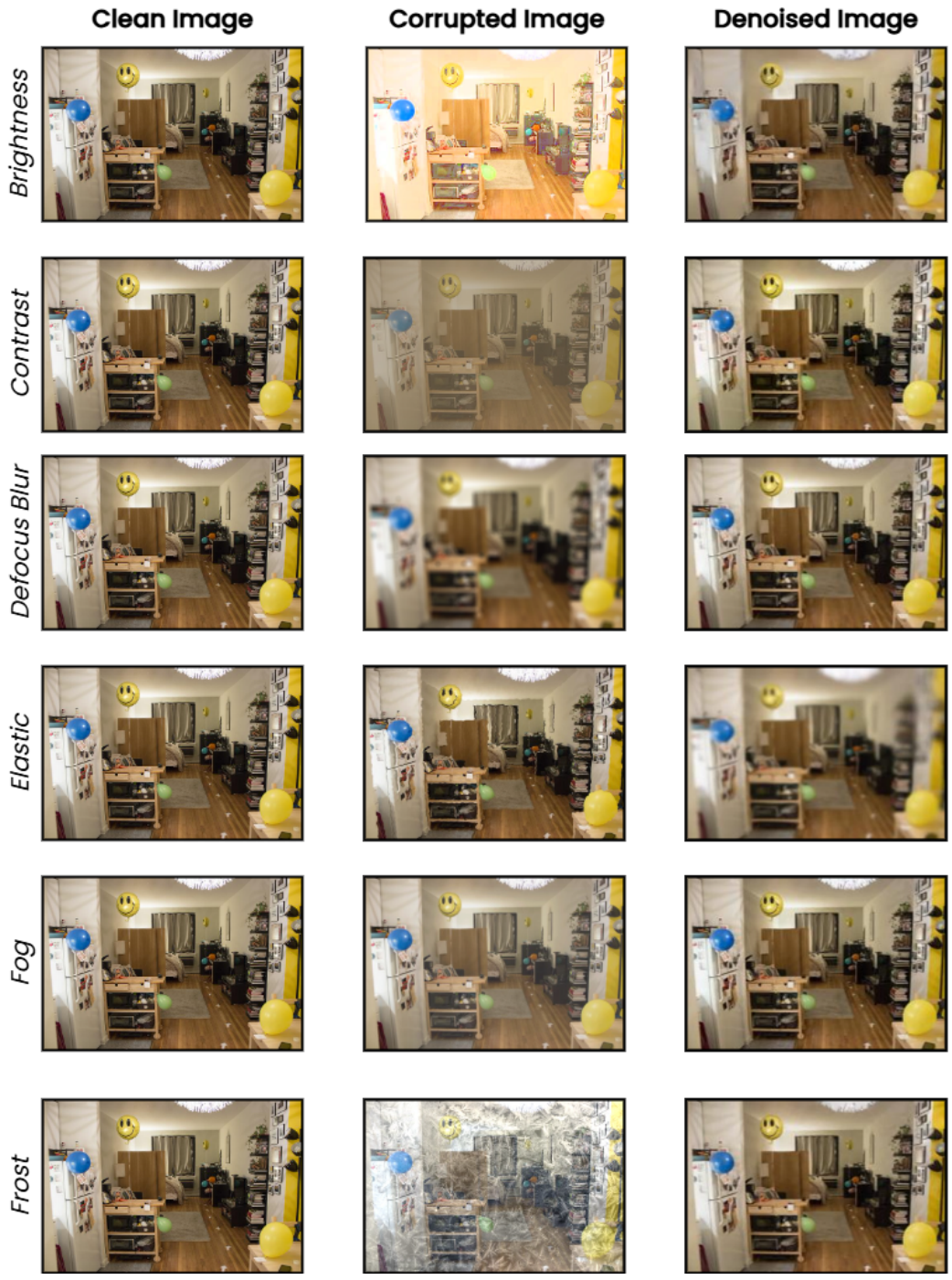


Figure 18. Clean, Corrupted, and Denoised images for the first set of 6 corruption types

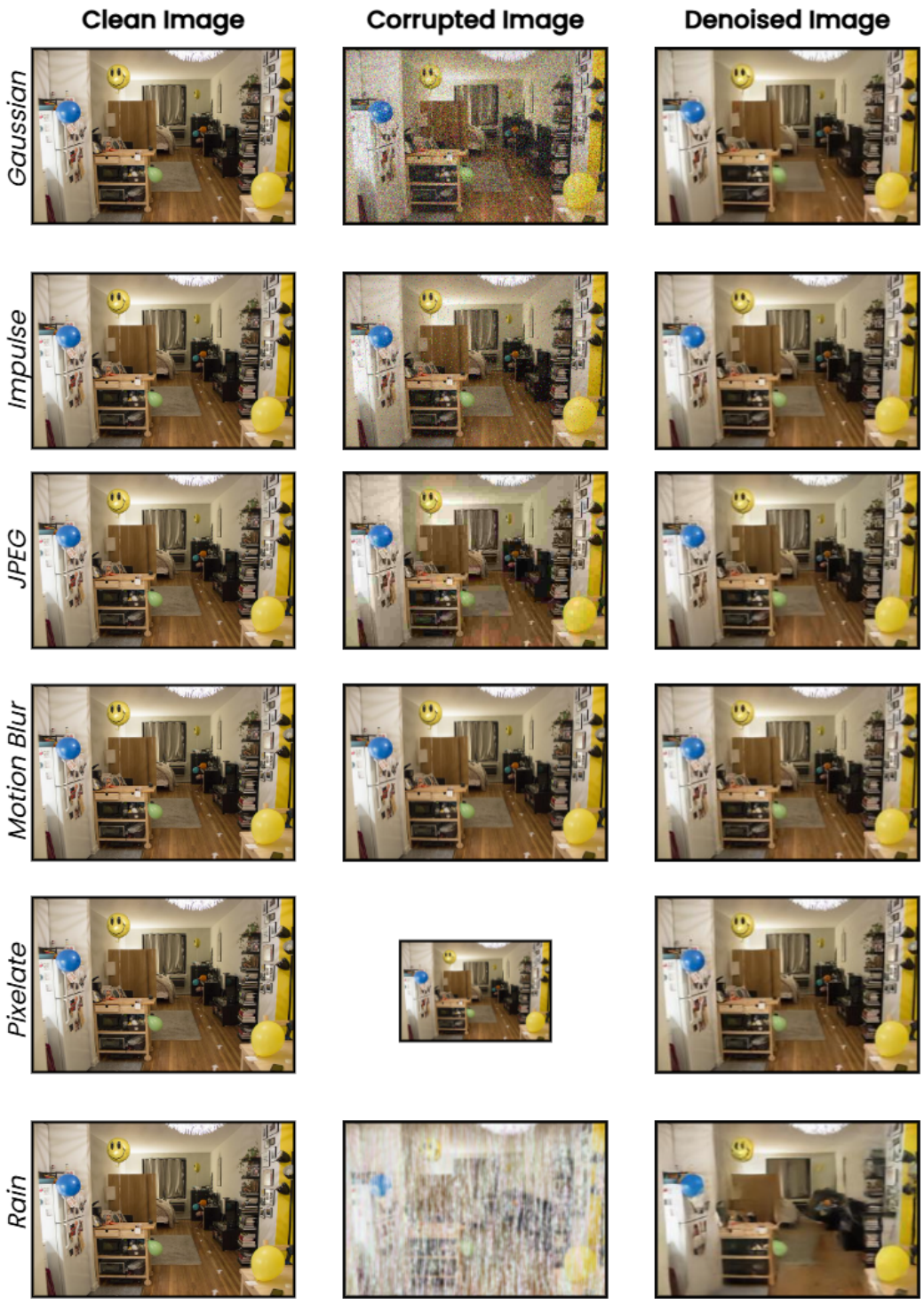


Figure 19. Clean, Corrupted, and Denoised images for the second set of 6 corruption types

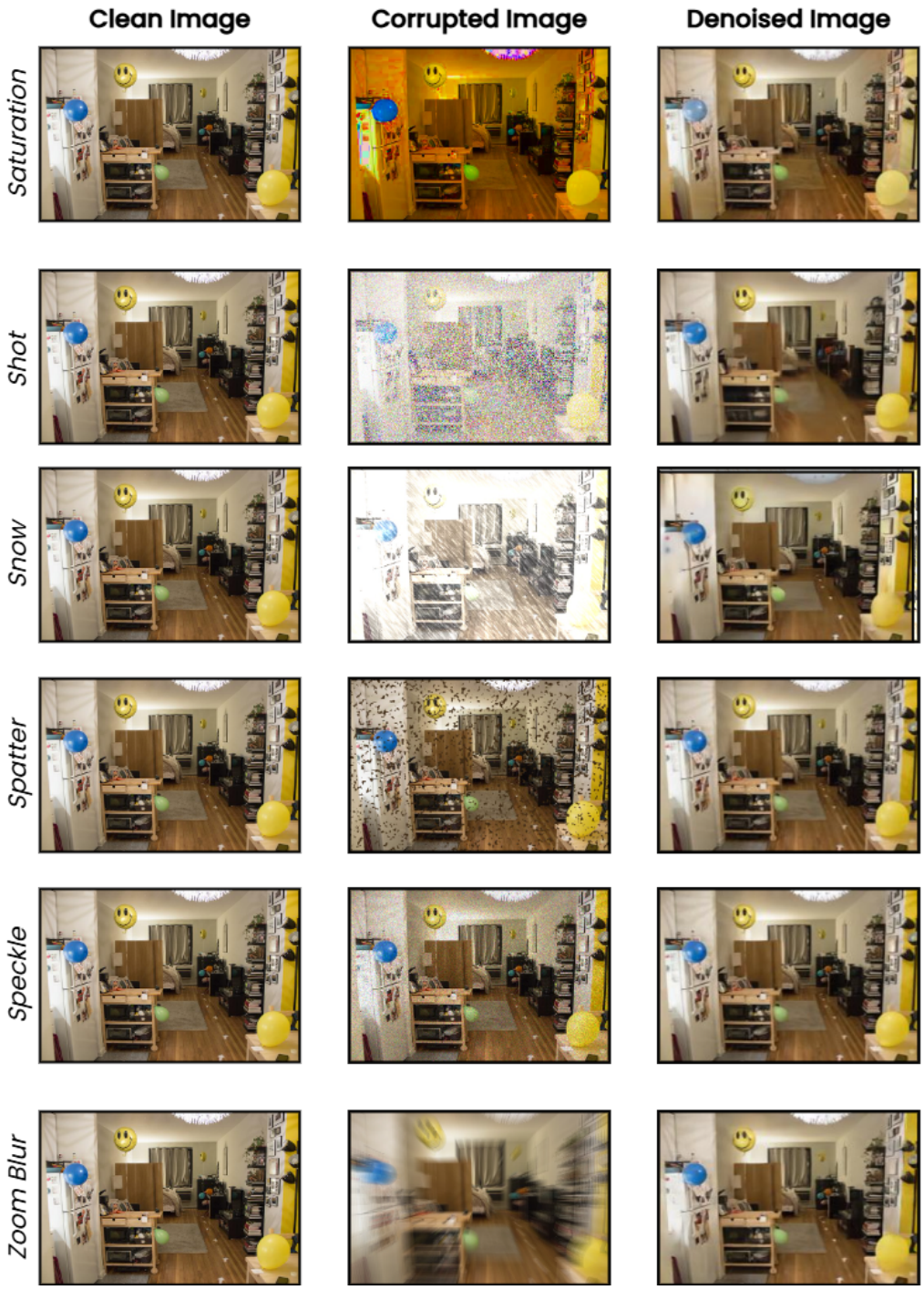


Figure 20. Clean, Corrupted, and Denoised images for the third set of 6 corruption types

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 the visual denoisers against different types of corruption effects.

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.