Grounding Degradations in Natural Language for All-In-One Video Restoration

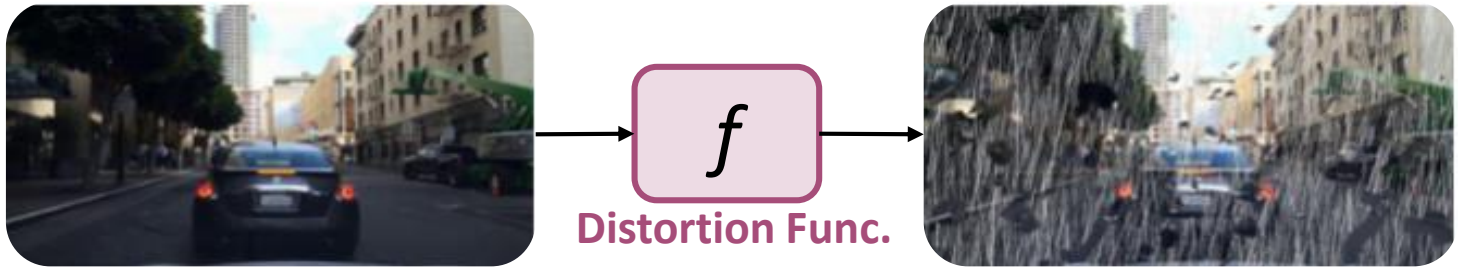
Muhammad Kamran Janjua*, Amirhosein Ghasemabadi*, Kunlin Zhang,
Mohammad Salameh, Chao Gao, Di Niu

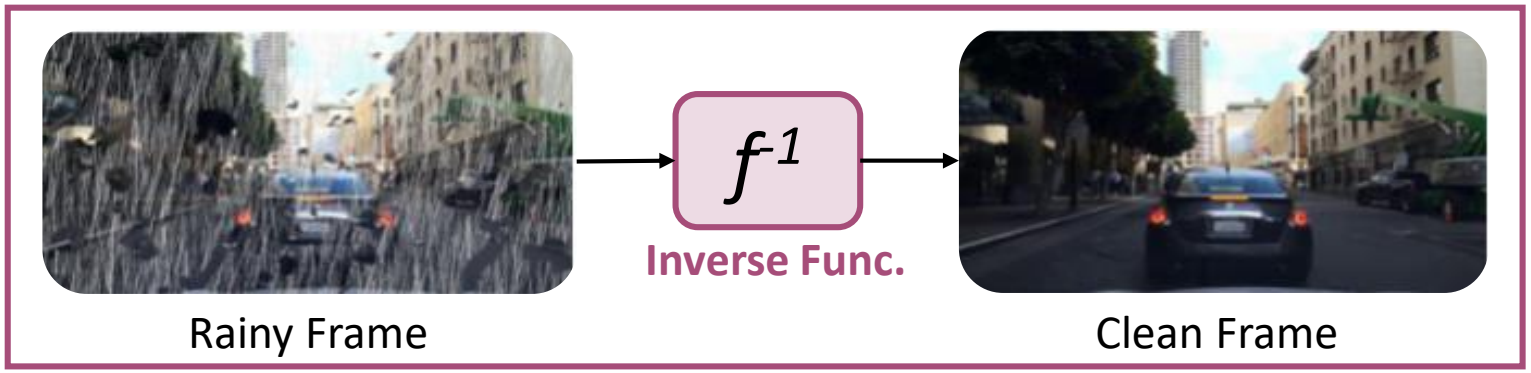
*Equal Contribution

IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2026

Restoration? What?

- Given a video degraded by any distortion (rain, snow, haze, blur, noise, etc.), we want to "*remove*" the degradation.





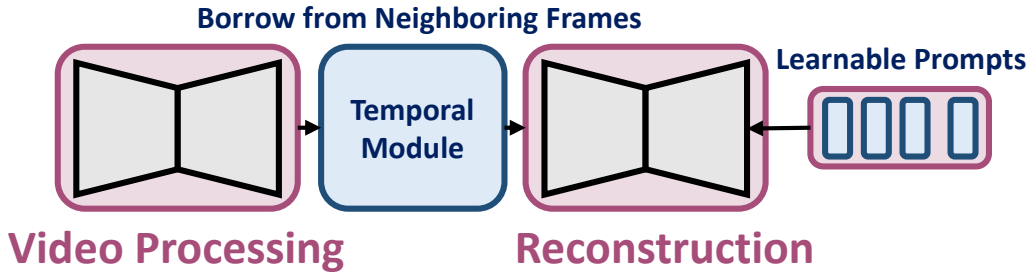
If we do not know *f*, then solution:
- Is Not Unique
- Many Possible Clean Frames Could Exist

→ Ill-Posed Problem

- If we know how the frame was degraded (function *f*), we can just inverse it.
- For all real-world scenarios, it is impossible to know *f,* so we learn it.
  - This is also known as "blind restoration"

But, What About Multiple Degradations?

- So far, only one network per degradation
  - A noise-removal network cannot remove blur, etc.
- To allow one network to restore multiple degradations, we need:
  - A way to condition the decoder (active some weights for noise, some for blur, etc.)
- How? Prompts!!
- Same overall three step design, but now we have learnable prompts in decoder (reconstruction)



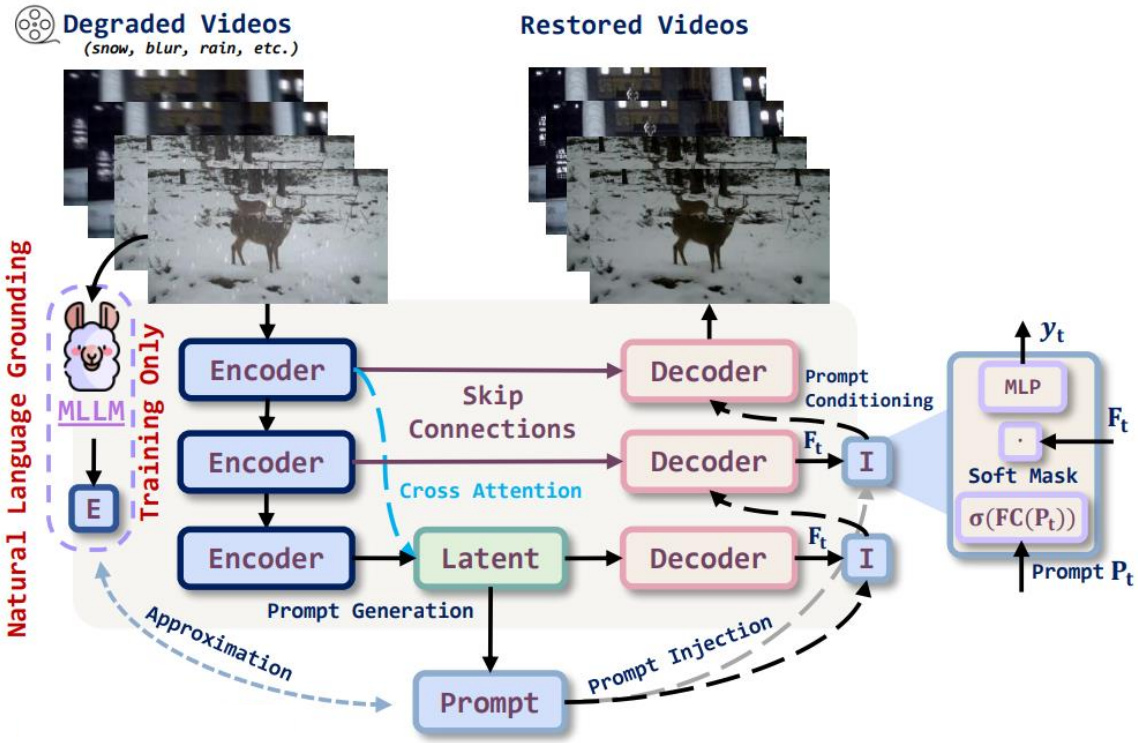
- These prompts interact with incoming features and at each stage, select different weights
- Prompt(s) for each degradation

What's Wrong with Literature?

- A few desirable qualities of these prompts:
  - They should be interpretable (we need to know what each prompt is) [1]
  - They should be independent of any external network [2]
- Interpretable
  - "Remove the noise from this frame" [instructions]
  - "This image/video contains blur induced by motion" [descriptions]
- Independent
  - Language needs to be encoded into vectors so that NNs can take input
  - If it's done in training, it needs to be done at inference
  - Independent means, no need for external encoders or language models in inference
- Turns out, these two qualities are mutually exclusive in literature

Grounding Degradations in Natural Language

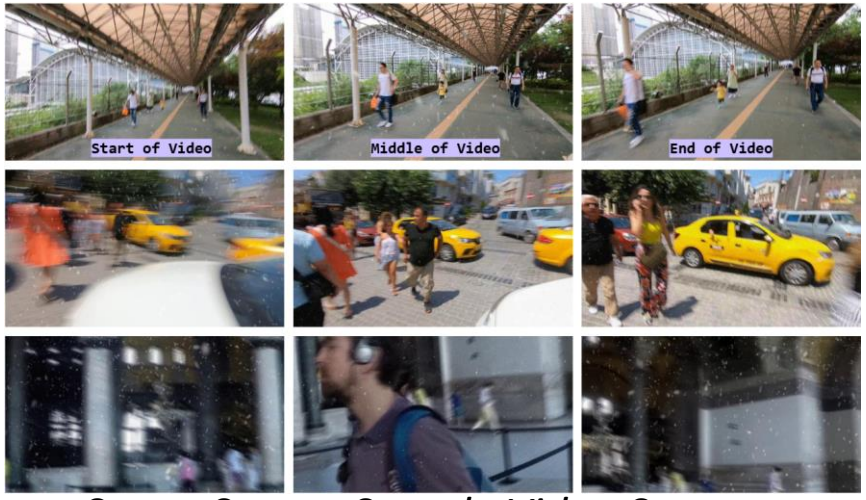
- Take a degradation-aware LLM, pass in each frame and generate degradation descriptions
- Use an encoder to encode each description into vectors as prompts for model [interpretable]
- Initialize a small MLP to approximate the language guidance during training [independent]



Each Frame's Degradation is Grounded in Language

Results – 3D, 4D, TUD, SnowyScenes

- We extend the prior time-varying TUD dataset [3] with a new all-in-one time-varying video restoration dataset – SnowyScenes
  - Varying snow intensity over time
- We consider: Three Degradations (3D), Four Degradations (4D), Time-Varying Unknown Degradations (TUD), and SnowyScenes



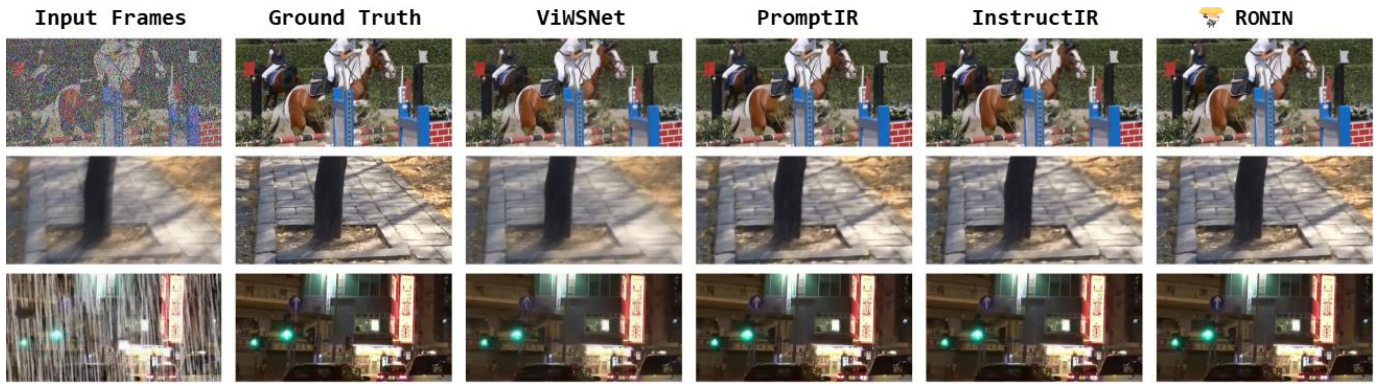
SnowyScenes Sample Video Scenes

Method	t = 6		t = 12		t = 24	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
AirNet [14]	23.41	0.62	23.51	0.64	23.44	0.61
AverNet [45]	22.34	0.58	21.93	0.58	21.88	0.55
InstructIR [5]	29.56	0.91	29.63	0.91	29.66	0.91
PromptIR [30]	29.72	0.91	29.79	0.91	29.81	0.91
ViWSNet [43]	27.22	0.87	27.27	0.87	27.33	0.87
RONIN	**30.21**	**0.92**	**30.28**	**0.92**	**30.27**	**0.92**

SnowyScenes Results

Setting	Method	Deblur (GoPro [26])		Denoise (DAVIS [29])		Derain (VRDS [39])		Desnow (RVSD [2])		Average	
		PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
3D	Restormer [44]	31.1653	0.9462	31.3816	0.9193	31.1068	0.9555			31.2179	0.9403
	InstructIR [5]	30.9331	0.9439	31.2521	0.9158	31.0966	0.9547			31.0939	0.9381
	PromptIR [30]	31.2833	0.9474	31.3529	0.9182	31.1776	0.9559	N/A		31.2713	0.9405
	ViWSNet [43]	27.8949	0.8949	29.9601	0.8863	28.5579	0.9234			27.6298	0.8250
	AverNet [45]	30.8064	0.9157	25.2306	0.4934	**32.8695**	0.9441			29.6355	0.7844
	RONIN	**32.7327**	**0.9605**	**31.6539**	**0.9220**	32.7224	**0.9656**			**32.3696**	**0.9493**
4D	Restormer [44]	29.6629	0.9286	31.0225	0.9117	29.8737	0.9437	25.9196	0.9263	29.1196	0.9275
	InstructIR [5]	29.4654	0.9260	31.0074	0.9125	29.8215	0.9442	24.8697	0.9163	28.7910	0.9247
	PromptIR [30]	29.7082	0.9296	31.0868	0.9130	30.2119	0.9481	**26.1032**	**0.9278**	29.2775	0.9296
	ViWSNet [43]	27.2592	0.8821	29.6782	0.8853	28.1486	0.9185	24.8427	0.9028	27.4806	0.8972
	RONIN	**30.7186**	**0.9417**	**31.2230**	**0.9160**	**31.1688**	**0.9544**	25.9538	0.9237	**29.7660**	**0.9339**

3D & 4D Tasks Results



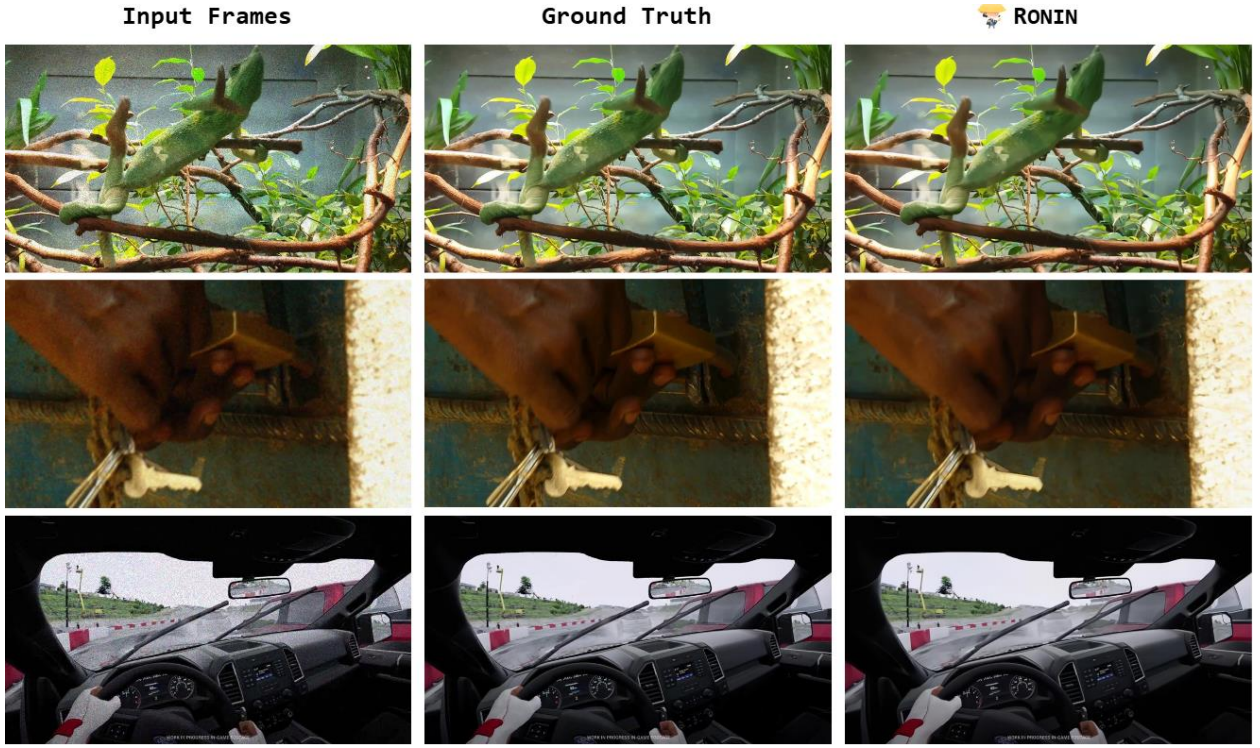
3D Visual Results

Thanks ☺

- More Results & Discussion in Paper
  - Read: https://arxiv.org/pdf/2507.14851
- References
  - [1] InstructIR: Conde, Marcos V., Gregor Geigle, and Radu Timofte. "Instructir: High-quality image restoration following human instructions." ECCV, 2024.
  - [2] PromptIR: Potlapalli, Vaishnav, et al. "Promptir: Prompting for all-in-one image restoration." NeurIPS, 2023.
  - [3] Zhao, Haiyu, et al. "AverNet: All-in-one video restoration for time-varying unknown degradations." NeurIPS, 2024.

Method	DAVIS [29]						Set8 [34]					
	t = 6		t = 12		t = 24		t = 6		t = 12		t = 24	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
WDiffusion [28]	31.74	0.8768	31.79	0.8784	31.92	0.8809	30.31	0.8784	30.02	0.8716	30.82	0.8746
TransWeather [36]	31.11	0.8684	31.13	0.8699	31.26	0.8741	29.24	0.8662	28.95	0.8565	29.15	0.8632
AirNet [14]	32.46	0.8873	32.46	0.8887	32.75	0.8929	30.71	0.8874	30.40	0.8806	31.16	0.8825
PromptIR [30]	31.18	0.8843	32.19	0.8867	32.45	0.8900	30.79	0.8903	30.43	0.8821	31.19	0.8847
EDVR [37]	28.70	0.7224	28.37	0.6991	29.07	0.7289	26.75	0.7259	26.94	0.7382	28.71	0.7675
BasicVSR++ [1]	33.22	0.9204	33.07	0.9180	33.32	0.9210	30.90	0.9048	30.52	0.8965	31.35	0.9011
ShiftNet [15]	33.09	0.9096	33.10	0.9113	33.34	0.9133	31.15	0.9027	30.82	0.8947	31.88	0.9000
RVRT [18]	33.99	0.9314	33.98	0.9311	34.10	0.9315	31.73	0.9192	31.39	0.9113	**32.47**	0.9178
AverNet [45]	**34.07**	0.9333	**34.09**	0.9339	**34.28**	0.9356	31.73	0.9219	31.47	0.9145	32.45	0.9189
RONIN	33.68	**0.9389**	33.82	**0.9408**	33.84	0.9411	**32.05**	**0.9504**	**32.11**	**0.9510**	32.20	**0.9523**

Time-Varying Degradation (TUD) Results



Time-Varying Degradation (TUD) Visual Results