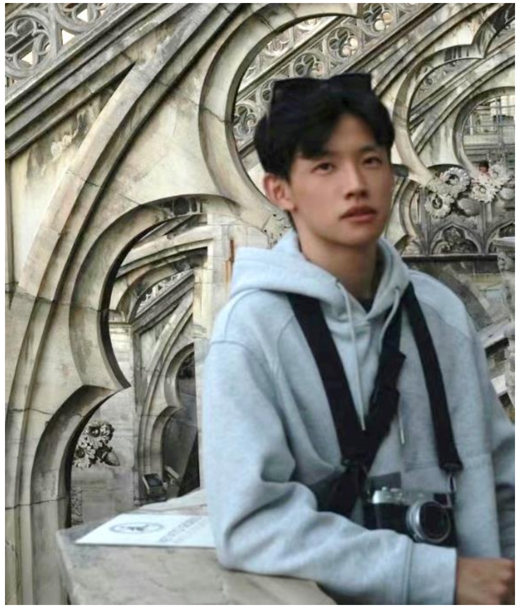
Genuine Knowledge from Practice: Diffusion Test-Time Adaptation for Video Adverse Weather Removal

CVPR 2024

Presenter: Jinyi Luo
2024.06.30

Authorship



Yijun Yang

PhD Candidate,

Hong Kong University of Science and Technology (Guangzhou)

- Education:
  - 2018-2022, Bachler, School of Computer Science and Technology, Shandong University
  - 2022-present, PhD candidate, Hong Kong University of Science and Technology (Guangzhou)
- Research interests:
  - Medical Image Analysis
  - Low-level vision
  - Generalizable AI

Authorship

Hongtao Wu

MPhil Student,

The Hong Kong University of Science and Technology (Guangzhou)

- Research interests:
    - Medical image analysis
    - Low-level vision
    - Video restoration
    - Computer vision

Authorship

Angelica I. Aviles-Rivero

Senior Research Associate,

DAMTP, University of Cambridge

- Education:
    - MS at the Department of Computer Engineering and Mathematics , Rovira i Virgili University
    - PhD at UPC – BarcelonaTech
- Experiences:
    - Visiting Researcher at George Washington University, 2016
- Research interests:
    - Computational Mathematics
    - Machine Learning
    - Computer Vision

Authorship

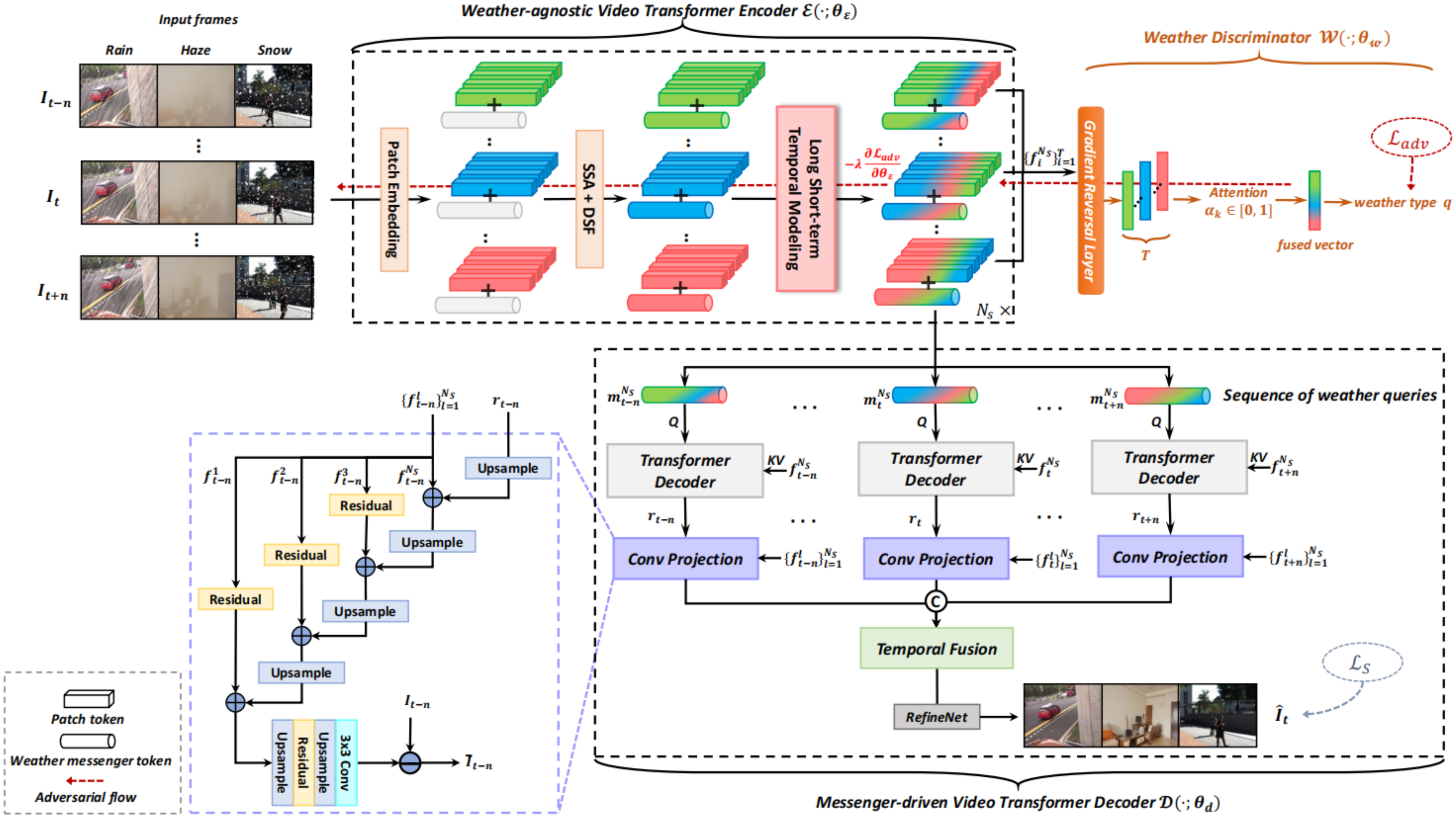
Lei Zhu

Assistant Professor, ROAS Thrust,

The Hong Kong University of Science and Technology (Guangzhou)

- Education:
  - PhD degree at Department of Computer Science and Engineering, the Chinese University of Hong Kong
  - Postdoctoral researcher at DAMTP, University of Cambridge
- Research interests:
  - Image restoration
  - Image enhancement
  - Image and video processing
  - Medical AI

Background: Previous Works: ViWS-Net



Background: Previous Works: ViWS-Net

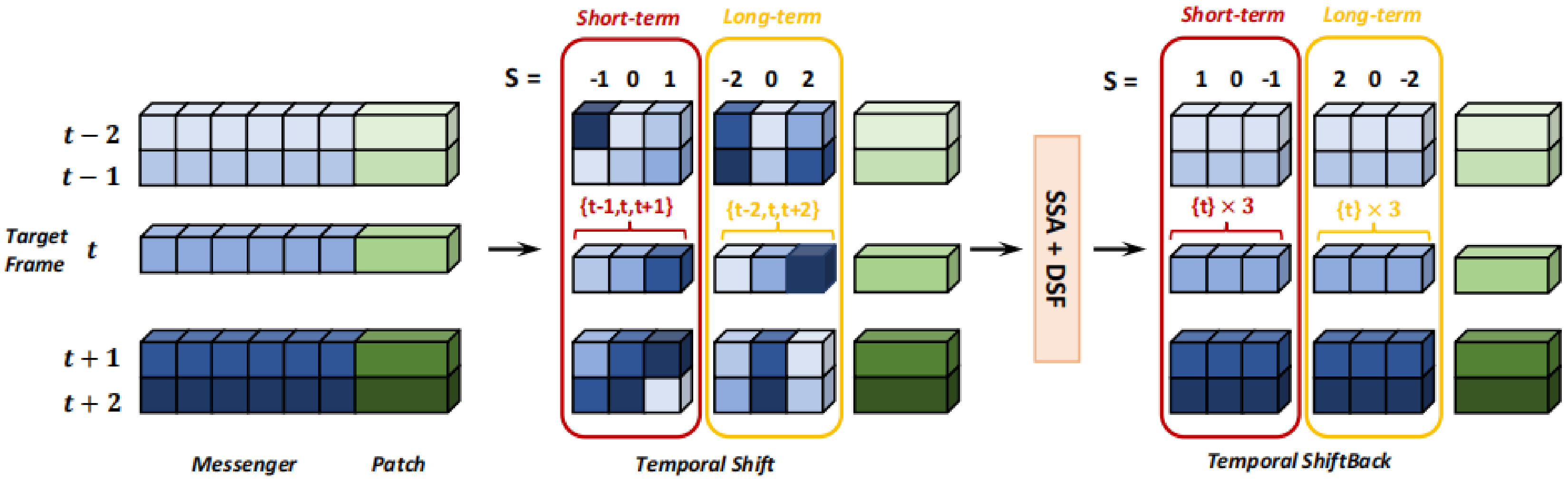
- Learnable weather embeddings as weather messenger tokens for each frame

$$\{m_i^0\}_{i=1}^T \in \mathbb{R}^{T\times M\times C} \qquad \{[f_i^0, m_i^0]\}_{i=1}^T \in \mathbb{R}^{T\times(\frac{HW}{P^2}+M)\times C}$$

- Encoder each patch and messenger token:

$$\{[f_i^l, m_i^l]\}_{i=1}^T = \{DSF^l(SSA^l([f_i^{l-1}, m_i^{l-1}]))\}_{i=1}^T.$$

- Lone-short term temporal modeling:



Background: Previous Works: ViWS-Net

PEKING UNIVERSITY 北京大學

- Compute the attention scores for patch tokens :

$$\alpha_i = \frac{\exp\{\mathbf{w}_1^T(tanh(\mathbf{w}_2\mathbf{v}_i^T)\cdot sigm(\mathbf{w}_3\mathbf{v}_i^T)\}}{\sum_{k=1}^T \exp\{\mathbf{w}_1^T(tanh(\mathbf{w}_2\mathbf{v}_k^T)\cdot sigm(\mathbf{w}_3\mathbf{v}_k^T)\}} \qquad \mathbf{v} = \sum_{i=1}^{T} \alpha_i \mathbf{v}_i,$$

- Adversarial loss on weather type:

$$\mathcal{L}_{adv} = \min_{\theta_w}\left(\lambda \max_{\theta_\varepsilon}(\sum_{q=1}^{Q}\sum_{i=1}^{N_q} q\log[\mathcal{W}(\mathcal{E}(\mathbf{V}_i^q)])\right)$$

- Supervised object losses:

$$\mathcal{L}_S = \mathcal{L}_{smoothL_1} + \gamma_1 \mathcal{L}_{perceptual}, \text{ with}$$

$$\mathcal{L}_{smoothL_1} = \begin{cases} 0.5(\hat{I}_t - B_t)^2, & if|\hat{I}_t - B_t| < 1 \\ |\hat{I}_t - B_t| - 0.5, & otherwise, \end{cases}$$

$$\mathcal{L}_{perceptual} = \mathcal{L}_{mse}(VGG_{3,8,15}(\hat{I}_t), VGG_{3,8,15}(B_t))$$

Background: Previous Works: VIWS-Net

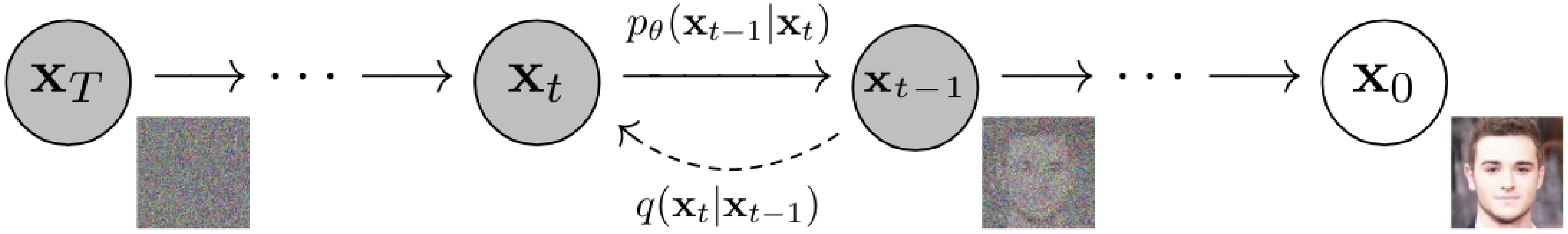
北京大學 PEKING UNIVERSITY

VIWS-Net:

- Introduce temporally-active weather messenger tokens that provide early temporal fusion
- Design a weather-suppression adversarial learning approach
- Maintains weather-invariant background information and suppresses weather-specific information

- Cannot adapt to unseen weather types
- Large model with high computational cost

Background: DDPM



$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \qquad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \qquad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

Apply multi-step Gaussian distributed noise
Train a network to predict parameters of the noise distribution
Use these parameters to sample during the reverse process to achieve generative denoising

Background: DDPM

Optimization: Applying ELBO:

$$\mathbb{E}\left[-\log p_\theta(\mathbf{x}_0)\right] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] = \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t\geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] =: L$$

Re-parameterization:

$$\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s) \qquad q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t\mathbf{I})$$

$$\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t \qquad \tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

Substitute these Gaussian-form distributions into the ELBO and simplify:

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\right\|^2\right]$$

Background: DDPM

北京大學 PEKING UNIVERSITY

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

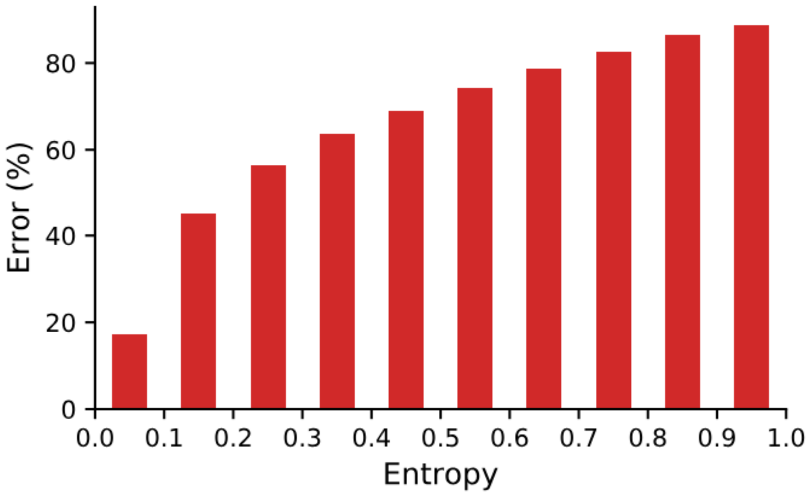
Algorithm 2 Sampling

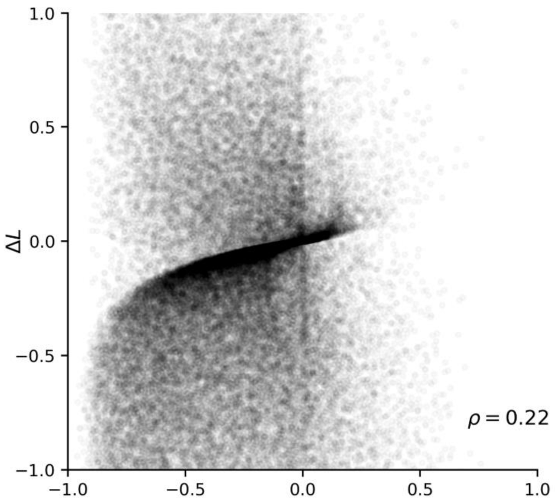
1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

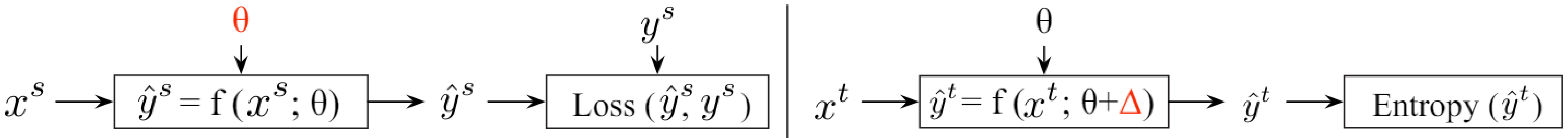
Ho et al. Denoising diffusion probabilistic models. Advances in neural information processing systems, 2020

Background: Test-Time Adaption

During test time, when the source label is inaccessible, there appears to be a positive correlation between the entropy of the generated results and the error rate.







(a) training

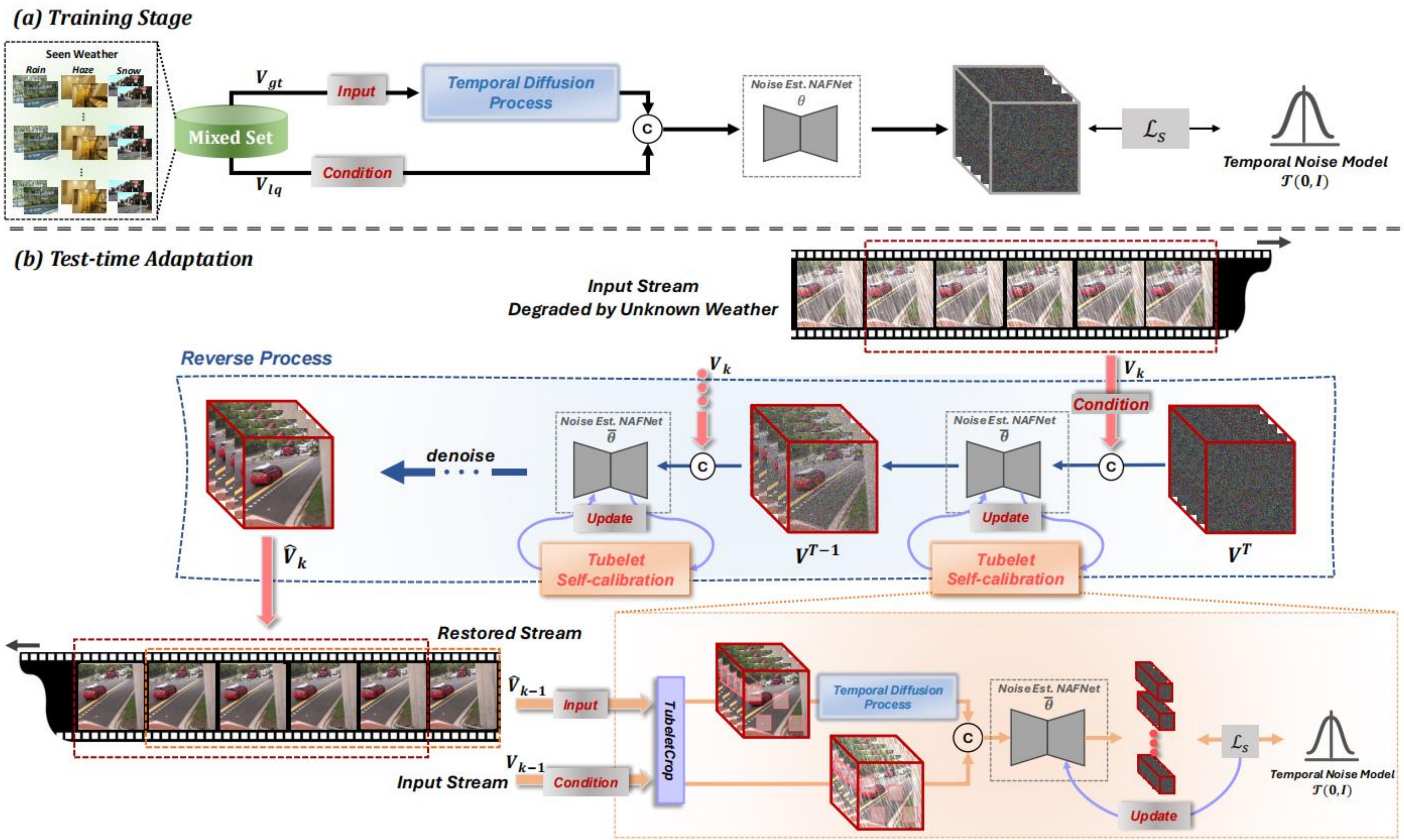
(b) fully test-time adaptation

Wang et al. Tent: Fully Test-Time Adaptation by Entropy Minimization, International Conference on Learning Representations, 2021

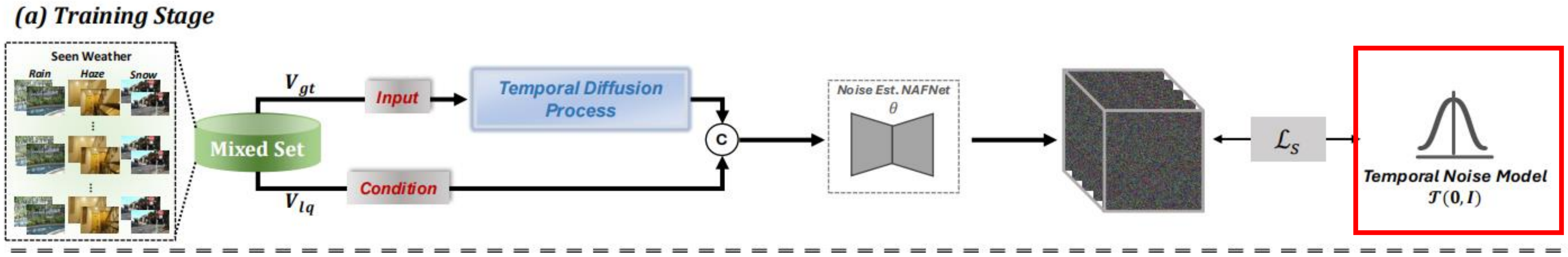
Task Description and Method Overview

- Task:
  - Restore high quality video clips from multiple weather degradations.
  - Adapt to unseen weather degradations, specifically during test-time
- Overview:
  - First diffusion-based adverse weather removal in videos
  - Leverages temporal redundancy information through a temporal noise model
  - Introduce test-time adaptation by incorporating a proxy task into the diffusion reverse process
- Performance: Achieved SoTA on multiple weather types with much less computation cost

Method Overview



Method: Training with Temporal Noise Model



For training: applying ARMA-formed temporal noise, substituting regular gaussian noise

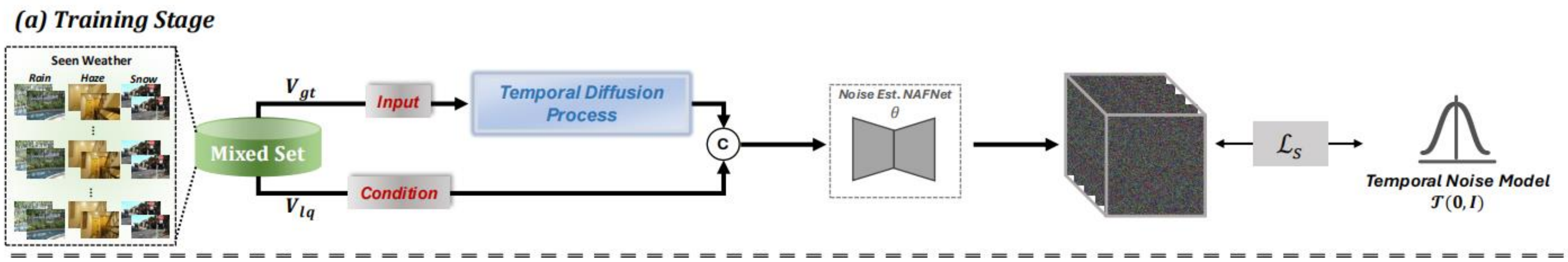
$$X_t = c + \varepsilon_t + \underbrace{\boxed{\sum_{i=1}^{p} \varphi_i X_{t-i}}}_{\textit{Auto Regression}} + \underbrace{\boxed{\sum_{j=1}^{q} \tau_j \varepsilon_{t-j}}}_{\textit{Moving Average}}$$

Taking both the next and previous frames into consideration

Constant c remain consistent with the mean of the variable:

$$\bar{\epsilon}_i = (1 - \varphi - \tau)\varepsilon_i + \varphi \frac{\bar{\epsilon}_{i-1} + \bar{\epsilon}_{i+1}}{2} + \tau \frac{\varepsilon_{i-1} + \varepsilon_{i+1}}{2}$$

Method: Training with Temporal Noise Model



Algorithm 1: Temporal Noise Model $\mathcal{T}(0, \mathbf{I})$

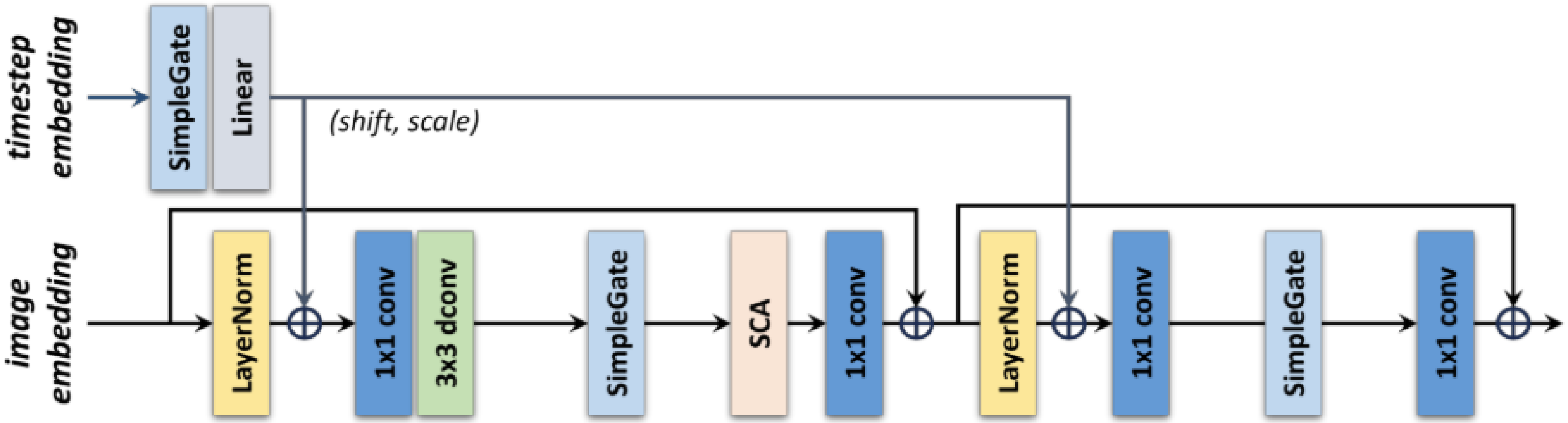
Input: K clips: $\{\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_K\}$ in a batch; The coefficients φ, τ

for *Clip* $k = 1, 2, ..., K$ ***parallelly*** **do**
    for *Frame* $i = 1, 2, ..., N_f$ ***parallelly*** **do**
        sample $\bar{\epsilon}_i, \varepsilon_i \in \mathbb{R}^{1\times c\times h\times w}$ from $\mathcal{N}(0, \mathbf{I})$
    end
    for *Frame* $i = 2, 3, ..., N_f$ ***sequentially*** **do**
        $\bar{\epsilon}_i = (1 - \varphi - \tau)\varepsilon_i + \varphi\bar{\epsilon}_{i-1} + \tau\varepsilon_{i-1}$
    end
    for *Frame* $i = N_f - 1, ..., 2, 1$ ***sequentially*** **do**
        $\bar{\epsilon}_i = (1 - \varphi - \tau)\varepsilon_i + \varphi\bar{\epsilon}_{i+1} + \tau\varepsilon_{i+1}$
    end
    $\bar{\epsilon} = normalize(\bar{\epsilon}) \Longrightarrow \bar{\epsilon} \sim \mathcal{T}(0, \mathbf{I})$
end
Return $\{\{\bar{\epsilon}_i | \bar{\epsilon}_i \in \mathbb{R}^{1\times c\times h\times w}\}_{i=1}^{N_f}\}_{k=1}^{K}$

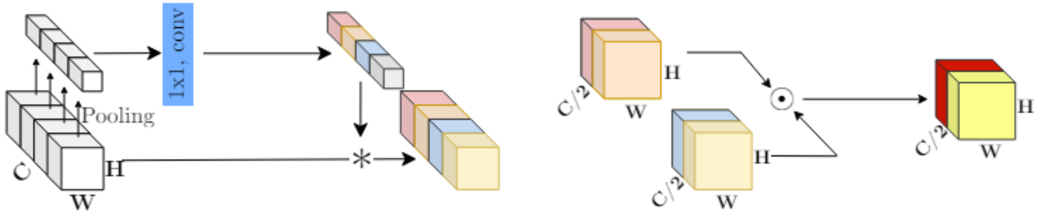
Optimize the NAFNet with conditioned ELBO loss in diffusion:

$$\mathcal{L}_S = \|\bar{\epsilon} - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\hat{\mathbf{V}} + \sqrt{1 - \bar{\alpha}_t}\bar{\epsilon}, \mathbf{V}, t)\|_1$$

Method: Noise Estimation model

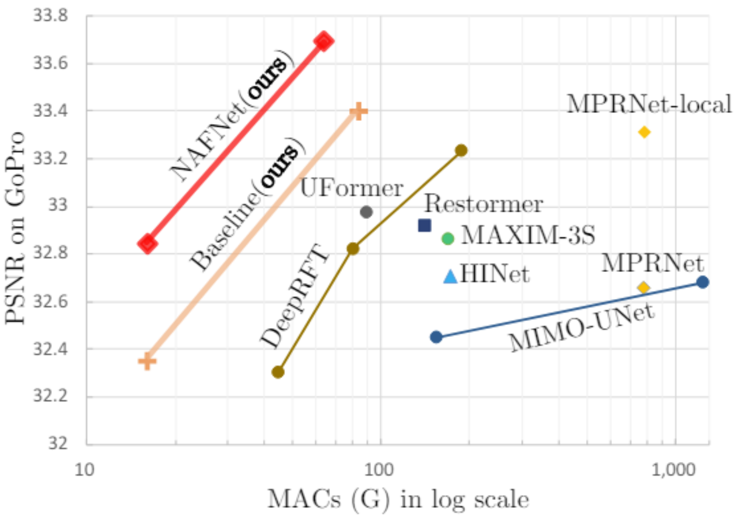


Simplified attention layers with low computational cost



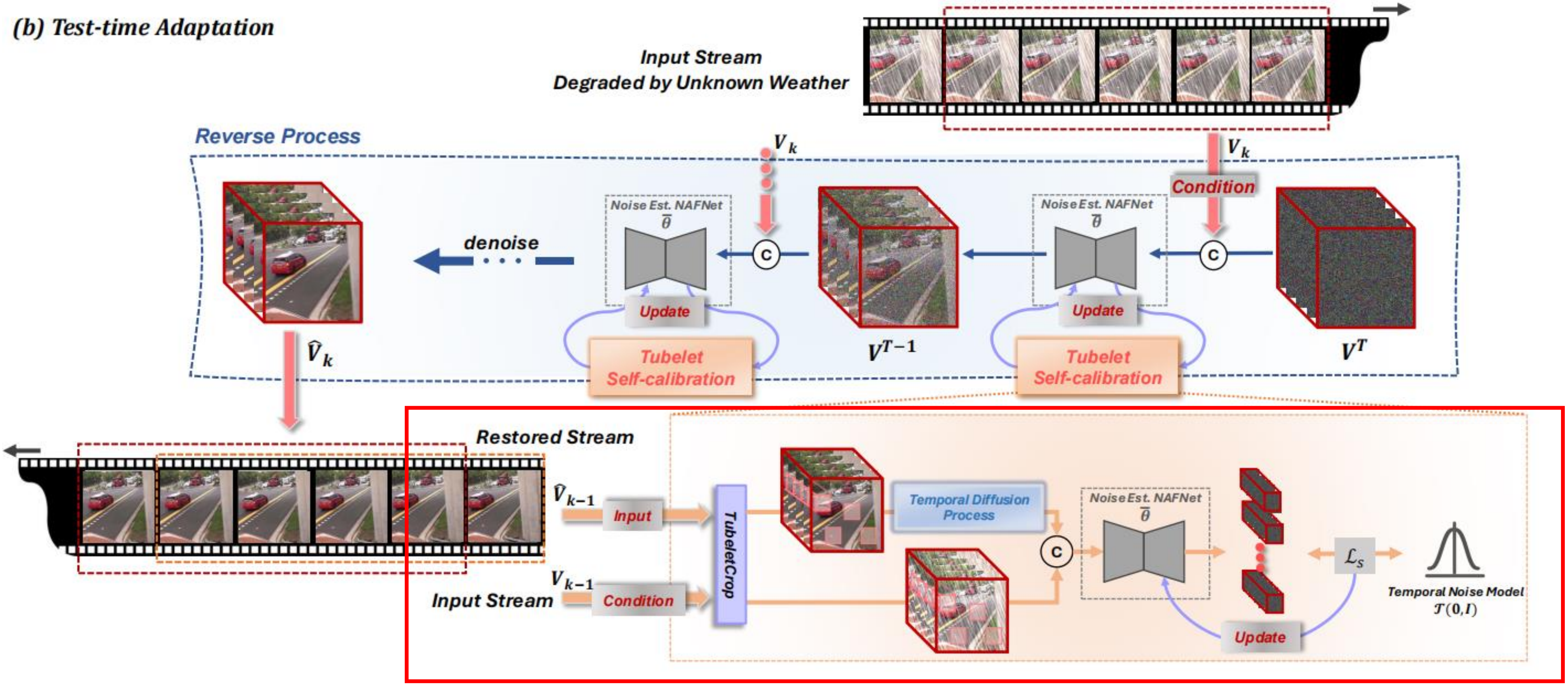
(b)Simplified Channel Attention

(c)Simple Gate



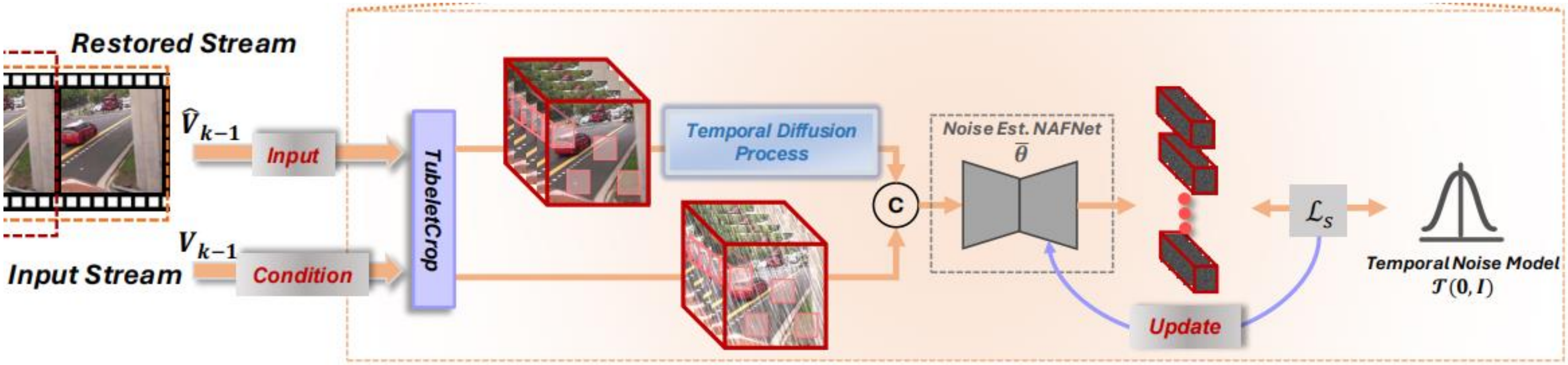
Chen et al. Simple Baselines for Image Restoration, The European Conference on Computer Vision , 2022

Method: Test-Time Adaption



Method: Test-Time Adaption

Introduce a proxy task: Tubelet Calibration to adapt the noise-estimation network to unseen weathers



Randomly crop tubelets from previously-generated clips, perform temporal noise estimation training on these tubelets and update the estimation network's parameters.

Method: Test-Time Adaption

PEKING UNIVERSITY

Algorithm 2: Diffusion Test-Time Adaptation to unknown weather. θ is the weight set of the trained network before adaptation, δ is the learning rate for online adaptation.

Input: K overlapped clips: $\{\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_K\}$ in one video stream

for *Clip* $k = 1, 2, ..., K$ ***sequentially*** **do**
    Initialize the network $\epsilon_{\bar{\theta}}$ with θ
    Initialize $\hat{\mathbf{V}}_k$ by Algorithm 1
    if $k = 1$ **then**
        for *step* $t = T, \ldots, 2, 1$ ***sequentially*** **do**
            $\hat{\mathbf{V}}_k = ddim(\mathbf{V}_k, \hat{\mathbf{V}}_k, \epsilon_{\bar{\theta}}, t)$
        end
    else
        $\mathbf{A}, \hat{\mathbf{A}} = TubeletCrop(\mathbf{V}_{k-1}, \hat{\mathbf{V}}_{k-1}, N_a)$
        for *step* $t = T, \ldots, 2, 1$ ***sequentially*** **do**
            Compute $\mathcal{L}_S(\mathbf{A}, \hat{\mathbf{A}}, \bar{\epsilon}, t)$ by Eq. (5)
            $\bar{\theta} = \bar{\theta} - \delta \nabla_{\bar{\theta}} \mathcal{L}_S$
            $\hat{\mathbf{V}}_k = ddim(\mathbf{V}_k, \hat{\mathbf{V}}_k, \epsilon_{\bar{\theta}}, t)$
        end
        $\hat{\mathbf{V}}_k = integrate(\hat{\mathbf{V}}_k, \hat{\mathbf{V}}_{k-1})$
    end
end
Return the restored clips $\{\hat{\mathbf{V}}_1, \hat{\mathbf{V}}_2, ..., \hat{\mathbf{V}}_K\}$

- Conditioned on the degraded frames, conduct temporal noise estimation on the previously-restored frames
- Enhances the consistency of generated frames under dynamic weather degradation
- Another possible explanation:
  Reduces the entropy of the generated video
- However still a test-time training domain adaption approach
- "Genuine knowledge" ?

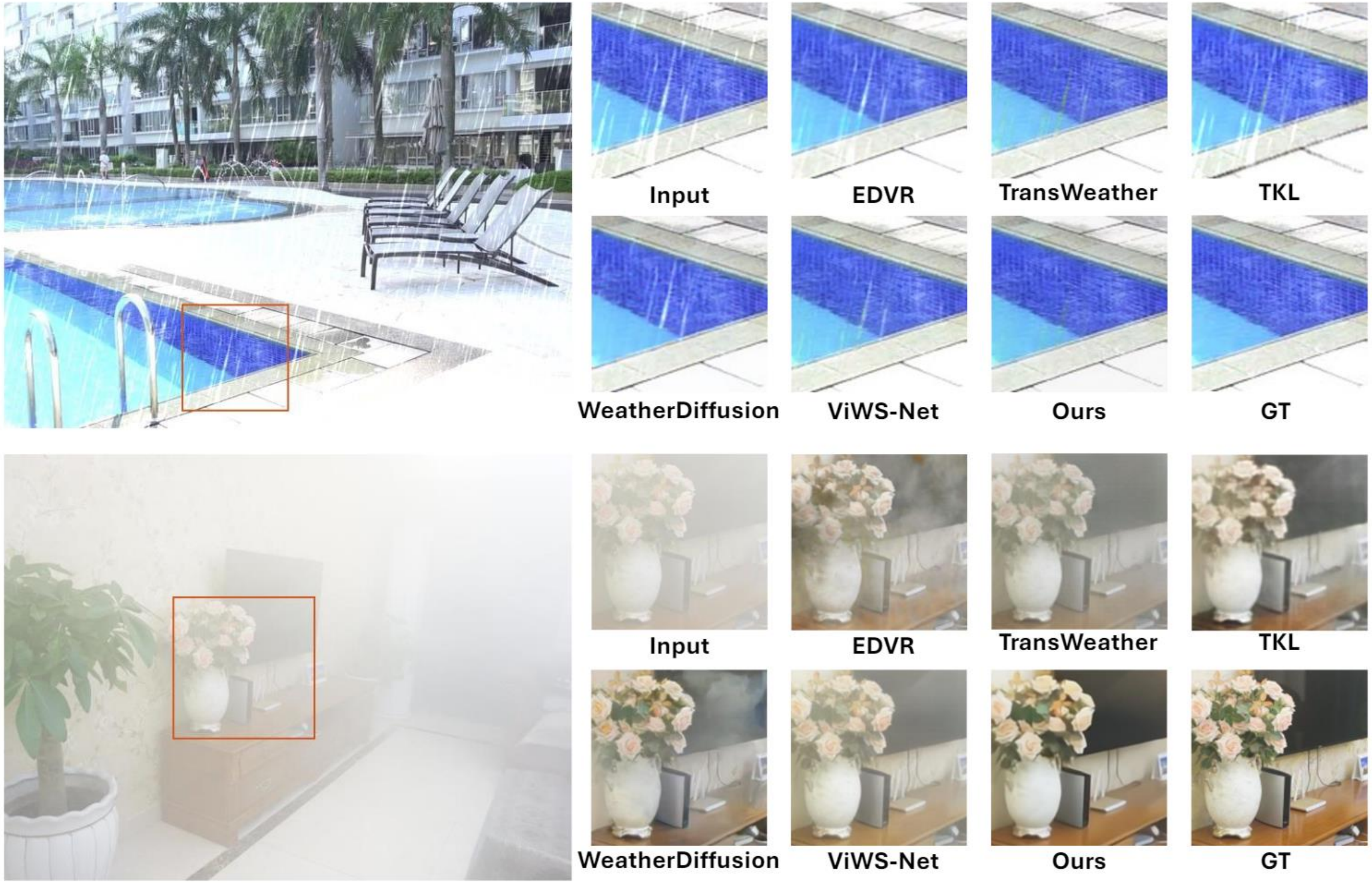
Experiments

Datasets

- Accessible weathers: Rain-Motion, REVIDE, Snow-KITTI
- Unseen weathers: VRDS, RVSD for out-of-distribution rain and snow
- Several real-world videos

Trained on RTX 4090s, 6.01s/ iter vs. 542.76s/ iter on WeatherDiffusion

Experiments: Results on Seen Weather Conditions

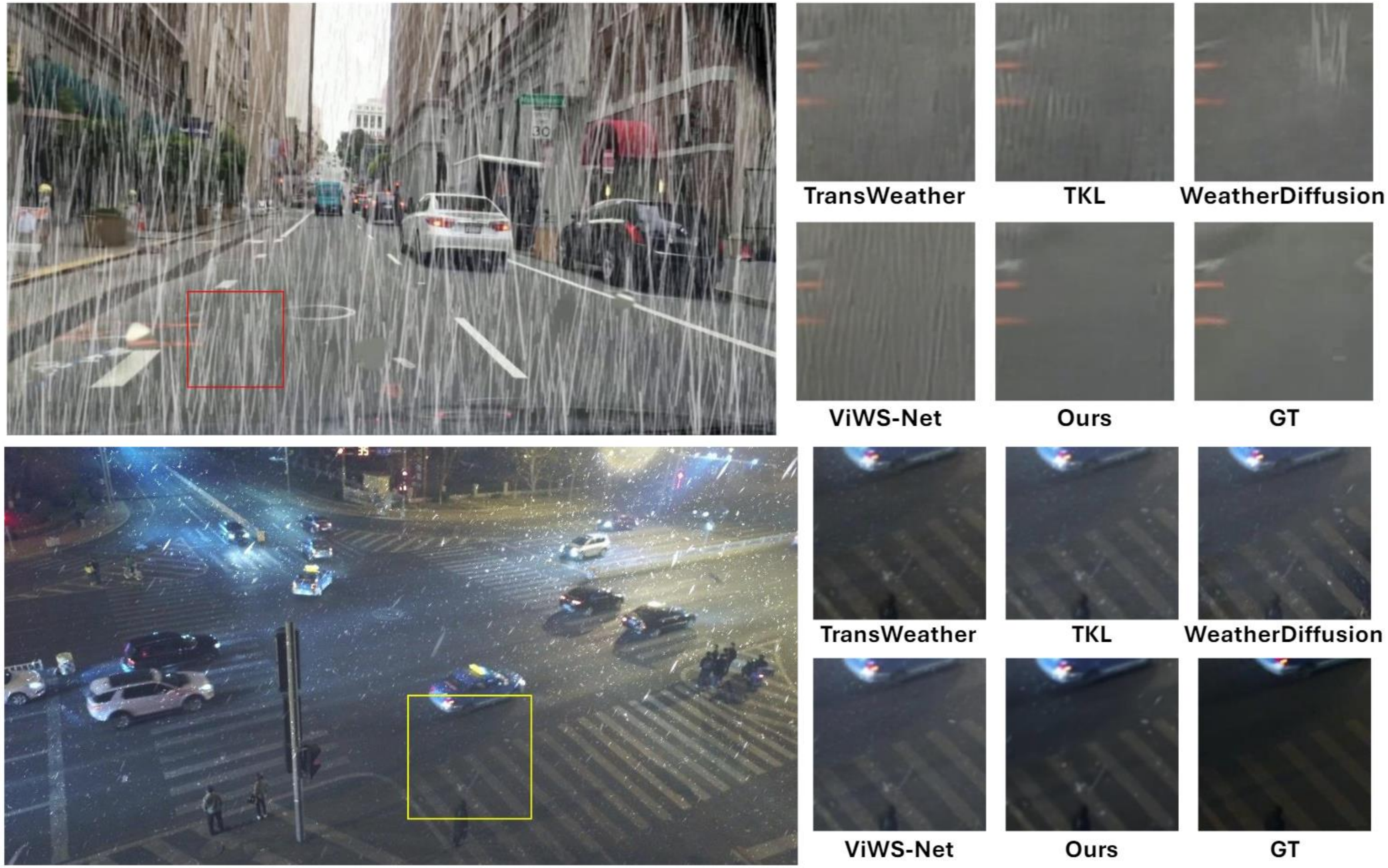


Experiments: Results on Seen Weather Conditions

Methods		Type	Source	Datasets									
				Original Weather		Rain		Haze		Snow		Average	
Derain	**PReNet** [35]	Image	CVPR'19	27.06	0.9077	26.80	0.8814	17.64	0.8030	28.57	0.9401	24.34	0.8748
	SLDNet [51]	Video	CVPR'20	20.31	0.6272	21.24	0.7129	16.21	0.7561	22.01	0.8550	19.82	0.7747
	S2VD [55]	Video	CVPR'21	24.09	0.7944	28.39	0.9006	19.65	0.8607	26.23	0.9190	24.76	0.8934
	RDD-Net [43]	Video	ECCV'22	31.82	0.9423	30.34	0.9300	18.36	0.8432	30.40	0.9560	26.37	0.9097
Dehaze	**GDN** [26]	Image	ICCV'19	19.69	0.8545	29.96	0.9370	19.01	0.8805	31.02	0.9518	26.66	0.9231
	MSBDN [15]	Image	CVPR'20	22.01	0.8759	26.70	0.9146	22.24	0.9047	27.07	0.9340	25.34	0.9178
	VDHNet [36]	Video	TIP'19	16.64	0.8133	29.87	0.9272	16.85	0.8214	29.53	0.9395	25.42	0.8960
	PM-Net [28]	Video	MM'22	23.83	0.8950	25.79	0.8880	23.57	0.9143	18.71	0.7881	22.69	0.8635
Desnow	**DesnowNet** [29]	Image	TIP'18	28.30	0.9530	25.19	0.8786	16.43	0.7902	27.56	0.9181	23.06	0.8623
	DDMSNET [58]	Image	TIP'21	32.55	0.9613	29.01	0.9188	19.50	0.8615	**32.43**	**0.9694**	26.98	0.9166
	HDCW-Net [10]	Image	ICCV'21	31.77	0.9542	28.10	0.9055	17.36	0.7921	31.05	0.9482	25.50	0.8819
	SMGARN [13]	Image	TCSVT'22	33.24	0.9721	27.78	0.9100	17.85	0.8075	32.34	0.9668	25.99	0.8948
Restoration	**MPRNet** [56]	Image	CVPR'21	——	——	28.22	0.9165	20.25	0.8934	30.95	0.9482	26.47	0.9194
	EDVR [44]	Video	CVPR'19	——	——	31.10	0.9371	19.67	0.8724	30.27	0.9440	27.01	0.9178
	RVRT [23]	Video	NIPS'22	——	——	30.11	0.9132	21.16	0.8949	26.78	0.8834	26.02	0.8972
	RTA [63]	Video	CVPR'22	——	——	30.12	0.9186	20.75	0.8915	29.79	0.9367	26.89	0.9156
Multi-Weather	**All-in-one** [21]	Image	CVPR'20	——	——	26.62	0.8948	20.88	0.9010	30.09	0.9431	25.86	0.9130
	UVRNet [19]	Image	TMM'22	——	——	22.31	0.7678	20.82	0.8575	24.71	0.8873	22.61	0.8375
	TransWeather [39]	Image	CVPR'22	——	——	26.82	0.9118	22.17	0.9025	28.87	0.9313	25.95	0.9152
	TKL [11]	Image	CVPR'22	——	——	26.73	0.8935	22.08	0.9044	31.35	0.9515	26.72	0.9165
	WeatherDiffusion [34]	Image	TPAMI'23	——	——	25.86	0.9125	20.10	0.8442	26.40	0.9113	24.12	0.8893
	WGWS-Net [64]	Image	CVPR'23	——	——	29,64	0.9310	17.71	0.8113	31.58	0.9528	26.31	0.9265
	ViWS-Net [52]	Video	ICCV'23	——	——	31.52	0.9433	24.51	**0.9187**	31.49	0.9562	29.17	0.9394
	Ours	Video	——	——	——	**32.43**	**0.9573**	**24.56**	0.9148	31.86	0.9640	**29.63**	**0.9453**

Capable of handling weathers with different physical characteristics such as rain and haze

Experiments: Results on Unseen Weather Conditions



New weather conditions, but no unseen types of degradation

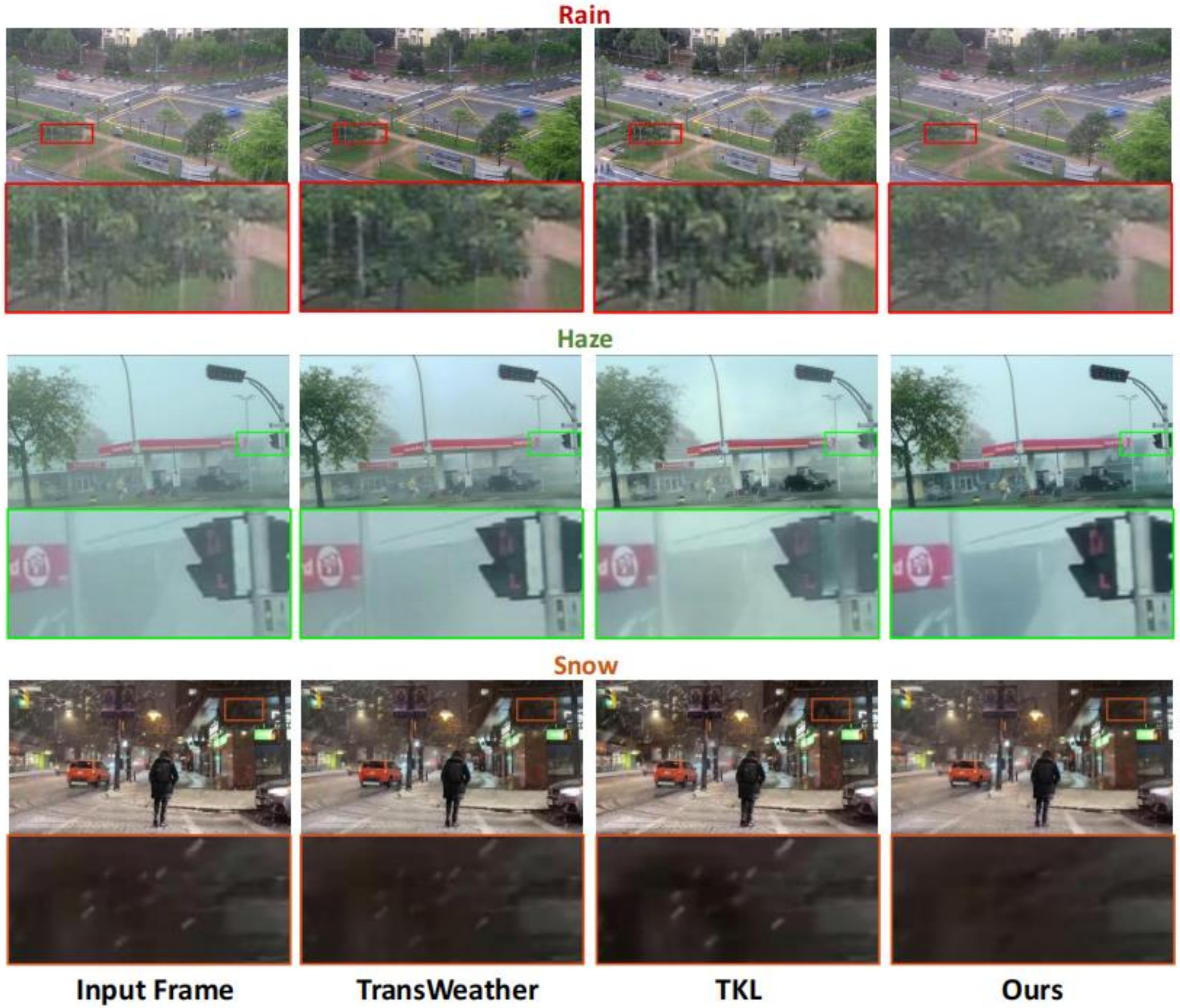
Exhibited better color restoration capability.

Experiments: Results on Unseen Weather Conditions

北京大學 PEKING UNIVERSITY

Method	VRDS [45]		RVSD [4]	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
All-in-one [21]	20.44	0.5944	19.79	0.7509
TransWeather [39]	21.36	<u>0.7136</u>	<u>20.25</u>	0.7514
TKL [11]	20.49	0.7003	19.71	0.7370
WeatherDiffusion [34]	20.73	0.6943	18.08	0.6588
ViWS-Net [52]	<u>21.57</u>	0.7094	19.83	<u>0.7590</u>
Diff-TTA (ours)	**22.57**	**0.7281**	**22.35**	**0.7719**

Experiments: Results on Real World Weather Conditions



Experiments

Visualization of the features from the last layer of the feature extractor

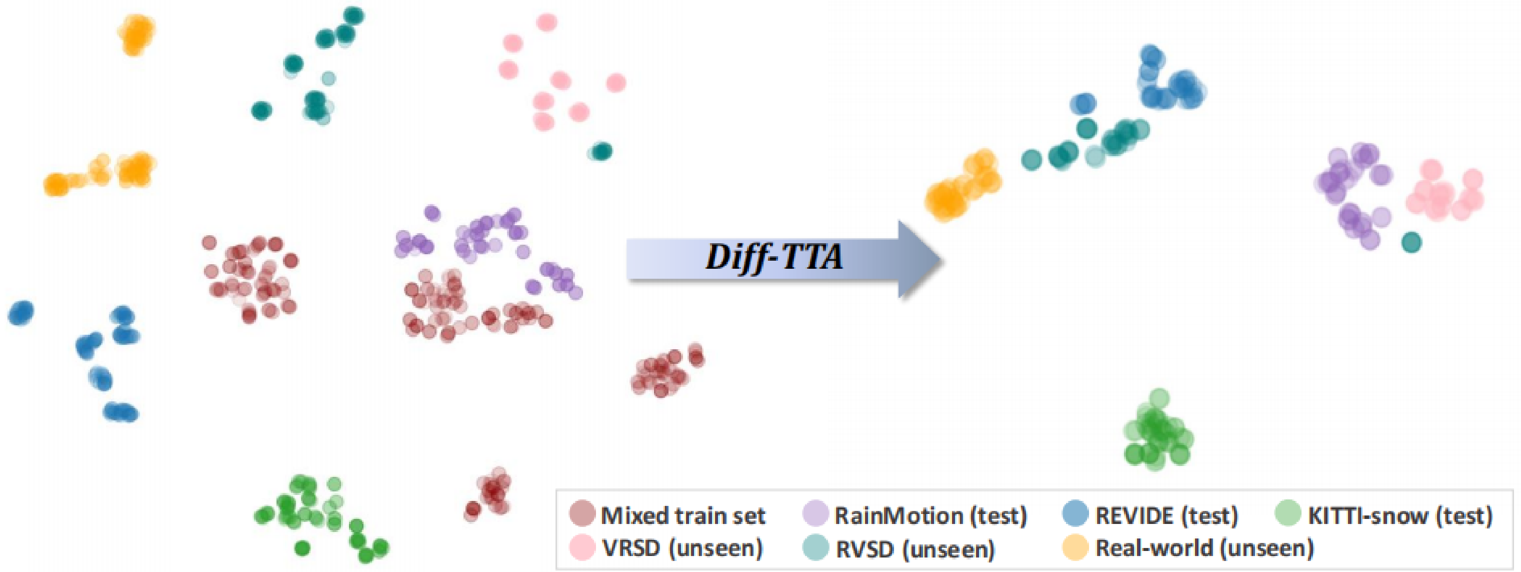
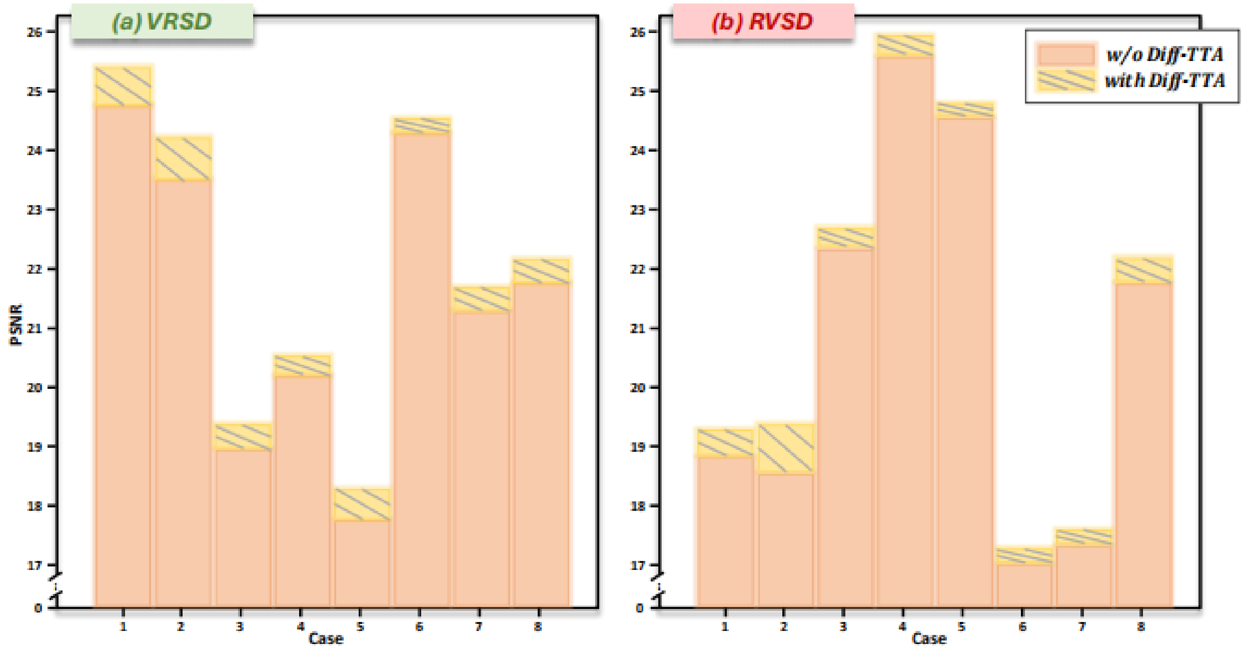


Figure 2. **Our Diff-TTA enables weather removal models to overcome unseen weather corruptions.** We use t-SNE [40] to visualize features from the last feature extractor layer of each dataset. Obviously, unseen data points tend to approximate the seen ones after adaptation, which means Diff-TTA can categorize unknown degradation into known distribution. (‘Real-world’ contains video clips simultaneously degraded by fog and snow.)

Experiments: Ablation

Combination	Component			Datasets							
	Diffusion Process	Temporal Noise	Diff-TTA	Rain		Haze		Snow		Average	
M1	-	-	-	29.07	0.9514	22.64	0.8930	28.79	0.9350	26.83	0.9264
M2	✓	-	-	31.55	0.9466	23.58	0.9041	30.48	0.9520	28.52	0.9342
M3	✓	✓	-	32.10	0.9530	24.31	0.9124	31.04	0.9621	29.15	0.9425
Ours	✓	✓	✓	**32.43**	**0.9573**	**24.56**	**0.9148**	**31.86**	**0.9640**	**29.63**	**0.9453**



Conclusion

- Introduced a diffusion-based adverse weather removal framework for videos
- Applied temporal noise model to substitute the regular gaussian noise to explore frame-correlated information
- Conduct temporal-diffusion process on the restored tubelets during test-time to adapt the noise estimation model to unseen weather degradations

- The Diff-TTA method works more as a domain adapter which decreases entropy
- The performance improvement on unseen weather types is limited

Thanks for listening!

Presenter: Jinyi Luo
2024.06.30