LoopMoEVR: Loop-Based Degradation-Aware Mixture-of-Experts for Unified UHD Video Restoration

Anonymous authors
Paper under double-blind review

Abstract

Recently, unified high-definition image restoration has attracted considerable attention; however, existing models tend to excessively increase their depth in pursuit of improved generalization, which often yields only limited gains. Meanwhile, loop-based learning paradigms have drawn widespread attention due to their low parameter counts and strong regression capability, as exemplified by GPT-6 and looped Transformers. In this paper, we introduce the loop learning paradigm to address restoration tasks that require cross-domain learning. Specifically, we propose LoopMoEVR, a loop-based mixture-of-experts model capable of handling degraded ultra-high-definition (UHD) inputs. First, a degradation-conditioned low-rank loop embedding is designed to construct input-dependent stage conditions. Second, a spatio-temporal iterative adaptive normalization module, termed IterAda3DN, is developed to fuse local features with global loop context, thereby performing position-wise affine modulation. Finally, the expert branches further integrate the attention-updated local and global video states with the loop conditions to generate dedicated modulation parameters, while an input-conditioned depth predictor adaptively configures the number of loop iterations. With only approximately 0.884M trainable parameters, the proposed model uniformly handles UHD video dehazing, deraining, denoising, and low-light enhancement tasks, achieving state-of-the-art restoration performance on both public benchmarks and real-world scenarios.

1 Introduction

Unified video restoration aims to address multiple degradations with a single model while exploiting cross-frame information to recover details and maintain temporal consistency. Extending unified restoration to ultra-high-definition (UHD) videos introduces further challenges. A key challenge is to increase restoration depth while retaining the ability to adapt to different degradations and evolving feature states.

For unified UHD video restoration, MoCRA (Wang et al., 2026b) introduces UHV-4K-AIO, built from 100 clean 4K clips. It allocates conditioning capacity between a coarse branch and a full-resolution refiner according to degradation frequency, enabling native 4K restoration without optical flow using approximately 3.6M parameters. However, its frame-wise conditions do not iteratively refine intermediate representations, its fixed inference depth cannot adapt to degradation severity, and it does not explicitly model global video states or cross-iteration context. Unified UHD restoration therefore still calls for progressive refinement and adaptive depth within a limited parameter budget.

The loop learning paradigm offers a way to address these challenges by progressively refining restoration results over multiple iterations. The Universal Transformer (Dehghani et al., 2019) repeatedly applies self-attention transformations along depth, supporting variable computation steps with a fixed parameter set. Subsequent studies examine the expressivity of looped computation and the role of iteration-dependent modulation (Saunshi et al., 2025; Xu & Sato, 2025). SMELT further investigates looped MoE models under matched resource budgets, demonstrating the effectiveness of looped computation for language modeling (Wang et al., 2026a). LoopMoE combines looped computation with sparse experts and generates modulation parameters from iteration position and current local hidden

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

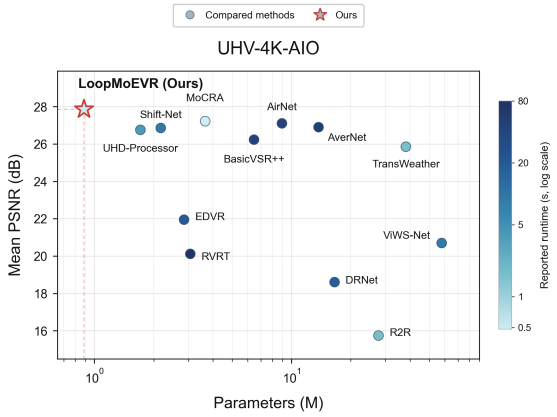

Figure 1: Mean PSNR, parameter count, and reported inference time on UHV-4K-AIO.



Figure 2: 4K video restoration on a smartphone.

states through IterAdaLN (Chen et al., 2026). However, its base iteration embeddings are determined by the iteration index without explicit input-degradation conditions, and its modulation pathways do not directly integrate global video states or cross-iteration context. In unified UHD video restoration, the same iteration position may correspond to different restoration needs under different degradations, while spatio-temporal variation within a video requires local updates to incorporate clip-level state and loop context. We therefore construct a conditioning mechanism driven jointly by degradation, iteration stage, and video state, while aligning expert modulation with local features updated by attention, to support iterative restoration across degradation conditions.

To this end, we propose LoopMoEVR, a loop-based mixture-of-experts model for unified UHD video restoration that incorporates degradation conditions and video-state information into iterative restoration. The model comprises input encoding, looped latent-feature updates, and output reconstruction. First, degradation-conditioned low-rank loop embeddings combine input degradation representations, a severity-related scalar, and iteration position to generate input-dependent conditions for each restoration stage. Second, the spatio-temporal iterative adaptive normalization module, IterAda3DN, integrates local spatio-temporal features, global video states, and cross-iteration context to modulate features at each position. The expert branch further uses local and global video states updated by attention to condition expert transformations. In addition, a depth predictor selects the number of iterations according to the input, adapting the restoration process to different computational needs. By repeatedly applying spatio-temporal processing layers, LoopMoEVR performs multi-stage restoration with approximately 0.884M parameters. At the same network width and four-stage execution depth, it uses 71.6% fewer parameters than a model with separate backbone parameters for each stage, with a PSNR difference of only 0.08 dB. This result shows that loop-based refinement achieves restoration quality close to that of stage-specific parameterization with a substantially more compact model, supporting our choice of a loop-based restoration architecture. Figure 1 compares the mean PSNR, parameter count, and reported inference time of LoopMoEVR and competing methods. Figure 2 shows a smartphone-based 4K video restoration example.

Our main contributions are as follows:

1. We propose LoopMoEVR for unified UHD video restoration, introducing the loop learning paradigm to progressively refine video features. By conditioning loop embeddings on input degradation representations, severity information, and iteration position, the model adapts feature updates to different degradation conditions and restoration stages.
2. We design IterAda3DN to integrate local spatio-temporal features with global loop context for position-wise affine modulation. Local and global video states updated by attention further provide dedicated modulation parameters for the expert branch, while an input-conditioned depth predictor assigns an appropriate loop depth to each input.
3. With approximately 0.884M trainable parameters, LoopMoEVR jointly addresses UHD video dehazing, deraining, denoising, and low-light enhancement. It achieves the highest

mean PSNR on UHV-4K-AIO and performs well in evaluations of time-varying degradations and temporal consistency.

2 Related Work

2.1 Unified Video Restoration

Unified video restoration combines temporal information with a shared model across degradations. AverNet (Zhao et al., 2024), UniWRV (Wan et al., 2026), and ORCANet (Li et al., 2026) use alignment, weather priors, and clip- or frame-level prompts, respectively; MoCRA (Wang et al., 2026b) targets native 4K restoration through frequency-aware, rank-one conditioning. In degradation-aware image restoration, AirNet (Li et al., 2022), PromptIR (Potlapalli et al., 2023), and UtilityIR (Chen & Pei, 2025) adapt features through learned representations or prompts; AdaIR (Cui et al., 2025) uses frequency cues, while MoCE-IR (Zamfir et al., 2025) and UniRestorer (Lin et al., 2026) study expert selection. UniLDiff (Cheng et al., 2026) incorporates degradation features into denoising, RONIN (Janjua et al., 2026) uses language-based degradation semantics, and video methods such as CDUN (Cheng et al., 2023) and LaverNet (Zhao et al., 2025) study adaptive updates and selective propagation. These approaches motivate input-aware restoration; our focus is degradation-conditioned, parameter-shared refinement of UHD video.

2.2 Looped Deep Learning

Looped models increase effective depth by reusing computation, as in the Universal Transformer (Dehghani et al., 2019), Huginn (Geiping et al., 2025), ELT (Goyal et al., 2026), and Looped World Models (Lu et al., 2026). Studies of loop expressivity (Saunshi et al., 2025) and timestep encoding (Xu & Sato, 2025) examine repeated transformations, while LoopFormer (Jeddi et al., 2026) conditions updates on iteration step size and Mixture-of-Recursions (Bae et al., 2025) routes token-level depth. LoopMoE (Chen et al., 2026) combines sparse experts with iteration- and local-state modulation; its expert branch already uses the attention-updated state. SMELT (Wang et al., 2026a) evaluates looped MoE under matched resource budgets. Building on this work, LoopMoEVR conditions loop embeddings on input degradation, incorporates global video state and cross-iteration context into modulation, and selects depth according to the input.

3 Method

3.1 Overall Framework

Figure 3 illustrates the overall mechanism of LoopMoEVR. Following the shared-computation setting of LoopMoE (Chen et al., 2026), given a degraded video clip $\mathbf{X}$, LoopMoEVR extracts latent features and degradation conditions, then iteratively updates video features through degradation-conditioned loop embeddings, spatio-temporal iterative adaptive normalization, and state-aware expert modulation. An input-conditioned depth predictor selects the iteration count from $\{3, 4, 5\}$, and a decoder fuses the final and shallow features to reconstruct $\hat{\mathbf{Y}}$.

3.2 Video Encoding and Degradation Condition Extraction

Given a degraded clip $\mathbf{X}$, the video encoder E downsamples only spatial dimensions to obtain latent features $\mathbf{h}_0$, while the shallow branch S retains full-resolution features $\mathbf{h}_s$. We then combine $\mathbf{h}_0$ with input channel statistics $\mu(\mathbf{X})$ to extract a degradation representation $\mathbf{z}$ and a severity-related scalar s:

$$\begin{aligned} \mathbf{h}_0 &= E(\mathbf{X}), \qquad \mathbf{h}_s = S(\mathbf{X}), \\ \mathbf{z} &= G_z([P(\mathbf{h}_0); \mu(\mathbf{X})]), \qquad s = \sigma(\mathbf{w}_s^\top \mathbf{z} + b_s), \end{aligned} \tag{1}$$

Here P globally averages spatio-temporal features, G_z maps degradation conditions, and σ is Sigmoid; $\mathbf{z}$ and s provide clip-level conditions for later iterations.

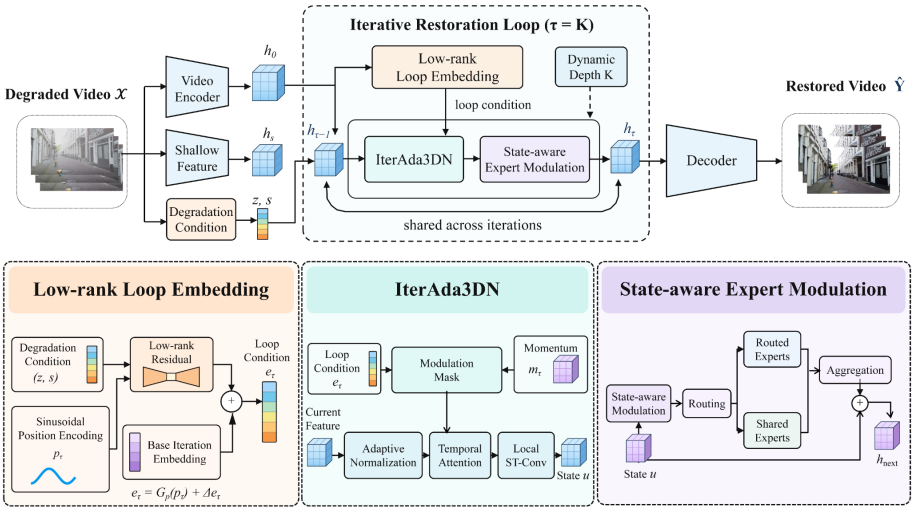

Figure 3: **Iterative restoration mechanism of LoopMoEVR.** Degradation-conditioned low-rank loop embedding guides IterAda3DN and state-aware expert modulation to progressively refine video features within a parameter-shared loop.

3.3 Degradation-Conditioned Low-Rank Loop Embedding

The degradation representation $\mathbf{z}$ and scalar s describe restoration needs but do not distinguish loop stages. We therefore combine them with the sinusoidal position encoding (Vaswani et al., 2017) $\mathbf{p}_\tau$ of iteration τ, correcting the base iteration embedding through a low-rank residual. First, we generate degradation and stage encodings $\mathbf{g}_{\text{deg}}$ and $\mathbf{g}_\tau$:

$$\begin{aligned} \mathbf{g}_{\text{deg}} &= \tanh(\mathbf{A}[\mathbf{z}; s] + \mathbf{a}) \,, \\ \mathbf{g}_\tau &= \tanh(\mathbf{B}\mathbf{p}_\tau + \mathbf{b}) \,, \end{aligned} \tag{2}$$

where $[\mathbf{z}; s]$ denotes concatenation, $\mathbf{A}, \mathbf{B}$ are projection matrices, and $\mathbf{a}, \mathbf{b}$ are biases. Both encodings lie in $\mathbb{R}^r$: $\mathbf{g}_{\text{deg}}$ remains constant within a clip, while $\mathbf{g}_\tau$ varies across stages.

Low-rank element-wise interactions have a precedent in bilinear pooling (Kim et al., 2017). Here, their element-wise interaction generates a conditional residual $\Delta\mathbf{e}_\tau$, which is added to the base embedding $G_p(\mathbf{p}_\tau)$ to obtain the current loop condition $\mathbf{e}_\tau$:

$$\begin{aligned} \Delta\mathbf{e}_\tau &= \mathbf{U}(\mathbf{g}_{\text{deg}} \odot \mathbf{g}_\tau) \,, \\ \mathbf{e}_\tau &= G_p(\mathbf{p}_\tau) + \Delta\mathbf{e}_\tau, \end{aligned} \tag{3}$$

where $\odot$ denotes element-wise multiplication, and $\mathbf{U} \in \mathbb{R}^{d_e \times r}$ maps the interaction into the embedding space. Choosing $r < d_e$ restricts the correction to a low-dimensional subspace, limiting additional parameters.

This interaction makes degradation corrections stage-dependent, providing conditions for different inputs and iterations. Zero-initializing $\mathbf{U}$ preserves the initial base embedding, while training learns degradation-dependent residuals. The next module combines $\mathbf{e}_\tau$ with video state and loop history for position-wise modulation.

3.4 Spatio-Temporal Iterative Adaptive Normalization

To adapt loop conditions to the current restoration state, IterAda3DN updates clip-level momentum $\mathbf{m}_\tau$ using the iteration input $\mathbf{h}_{\tau-1}$, loop embedding $\mathbf{e}_\tau$, and historical state:

$$\begin{aligned} \mathbf{m}_0 &= G_0([P(\mathbf{h}_0); \mathbf{z}]) \,, \\ \mathbf{m}_\tau &= 0.7\, G_m([P(\mathbf{h}_{\tau-1}); \mathbf{e}_\tau; \mathbf{m}_{\tau-1}]) + 0.3\, \mathbf{m}_{\tau-1}, \end{aligned} \tag{4}$$

where G_0, G_m initialize and update momentum. The state $\mathbf{m}_\tau$ propagates cross-iteration context and is reinitialized for each clip.

Using $\mathbf{m}_\tau$ and current features, we generate channel and spatio-temporal factors $\mathbf{M}_c, \mathbf{M}_{st}$, combining them into a modulation mask $\mathbf{M}_\tau$:

$$\begin{aligned} \mathbf{M}_c &= 2\sigma(\mathbf{W}_c\mathbf{m}_\tau + \mathbf{b}_c), \\ \mathbf{M}_{st} &= 2\sigma\big(\mathbf{W}_{st}\mathbf{h}_{\tau-1} + \mathrm{br}(\mathbf{w}_b^\top \mathbf{m}_\tau + b_b)\big), \\ \mathbf{M}_\tau &= \mathbf{M}_c \odot \mathbf{M}_{st}, \end{aligned} \tag{5}$$

where br denotes broadcasting. Zero-initialized projections yield an initial mask of 1, which each LoopLayer within the iteration applies to its current features.

Conditional affine modulation has a precedent in FiLM (Perez et al., 2018); here both local spatio-temporal features and global video context determine the parameters. For branch input $\mathbf{v}$, we combine its global statistics with loop information as $\mathbf{c} = [P(\mathbf{v}); \mathbf{e}_\tau; \mathbf{m}_\tau]$, then incorporate local features to generate scales $\boldsymbol{\gamma}_b$ and shifts $\boldsymbol{\beta}_b$:

$$\begin{aligned} f_b(\mathbf{v}, \mathbf{c}) &= \mathrm{GELU}\Big(\mathbf{W}_{\text{loc}}^{(b)}\mathbf{v} + \mathrm{br}(\mathbf{W}_{\text{glob}}^{(b)}\mathbf{c})\Big), \\ [\boldsymbol{\gamma}_b; \boldsymbol{\beta}_b] &= \tanh\Big(\mathbf{W}_{\text{aff}}^{(b)} f_b(\mathbf{v}, \mathbf{c})\Big), \end{aligned} \tag{6}$$

where $b = 1, 2$ denote the attention and expert branches. Local mappings retain spatio-temporal variation, while global mappings introduce video state and cross-iteration context.

The generated $\boldsymbol{\gamma}_b, \boldsymbol{\beta}_b$ apply position-wise channel normalization and affine modulation to $\mathbf{v}$:

$$\mathcal{N}(\mathbf{v}; \boldsymbol{\gamma}_b, \boldsymbol{\beta}_b) = (1 + \boldsymbol{\gamma}_b) \odot \frac{\mathbf{v}}{\sqrt{\frac{1}{d}\sum_{j=1}^{d} v_j^2 + \varepsilon}} + \boldsymbol{\beta}_b, \tag{7}$$

where v_j is the j-th channel value at the current position, d is the channel count, and ε ensures stability. Normalization uses the channel root mean square (Zhang & Sennrich, 2019), with affine parameters varying across spatio-temporal positions.

For current layer features $\mathbf{h}$, we first apply the mask to obtain $\tilde{\mathbf{h}} = \mathbf{M}_\tau \odot \mathbf{h}$, then produce an updated state $\mathbf{u}$ through the attention branch:

$$\mathbf{u} = \mathbf{h} + \boldsymbol{\alpha} \odot R_1\big(\mathcal{A}\big(\mathcal{N}(\tilde{\mathbf{h}}; \boldsymbol{\gamma}_1, \boldsymbol{\beta}_1)\big)\big), \tag{8}$$

where $\boldsymbol{\alpha}$ is a zero-initialized position-wise gate generated from modulation features, and R_1 is channel normalization with learnable scaling. The operator $\mathcal{A}$ combines frame-level temporal attention with local spatio-temporal convolution.

3.5 State-Aware Expert Modulation

Using the attention-updated state $\mathbf{u}$, loop embedding, and momentum, the expert branch forms $\mathbf{c}_2$ and obtains expert input $\mathbf{v}_2$ through affine modulation:

$$\begin{aligned} \mathbf{c}_2 &= [P(\mathbf{u}); \mathbf{e}_\tau; \mathbf{m}_\tau], \\ (\boldsymbol{\gamma}_2, \boldsymbol{\beta}_2) &= G_2(\mathbf{u}, \mathbf{c}_2), \\ \mathbf{v}_2 &= \mathcal{N}(\mathbf{u}; \boldsymbol{\gamma}_2, \boldsymbol{\beta}_2), \end{aligned} \tag{9}$$

Here G_2 is independent of the attention branch, so expert modulation uses the updated state.

Pooled features $P(\mathbf{v}_2)$ and momentum $\mathbf{m}_\tau$ yield routing scores $\mathbf{q}$ that select k_e experts in $\mathcal{S}$:

$$\mathbf{q} = \mathrm{nrm}(\mathbf{W}_x P(\mathbf{v}_2)) + \tanh(\rho)\,\mathrm{nrm}(\mathbf{W}_m \mathbf{m}_\tau),$$
$$\mathcal{S} = \mathrm{TopK}_{k_e}(\mathbf{q}), \qquad \pi_i = \frac{\exp(q_i)}{\sum_{j\in\mathcal{S}} \exp(q_j)}, \tag{10}$$

where nrm is root mean square normalization, ρ is a scalar gate, and π_i is the normalized weight of a selected expert. This top-k_e selection follows sparse expert routing (Shazeer et al., 2017). All positions within a clip use the same expert set.

Shared experts have a precedent in DeepSeekMoE (Dai et al., 2024). Outputs from routed experts E_i and shared experts are aggregated as $\mathcal{E}$ to update the features:

$$\mathcal{E}(\mathbf{v}_2, \mathbf{m}_\tau) = \sum_{i\in\mathcal{S}} \pi_i E_i(\mathbf{v}_2) + \frac{1}{N_s} \sum_{j=1}^{N_s} E_j^{\text{shared}}(\mathbf{v}_2),$$
$$\mathbf{h}_{\text{next}} = \mathbf{u} + R_2(\mathcal{E}(\mathbf{v}_2, \mathbf{m}_\tau)), \tag{11}$$

where E_i comprises temporal convolution, spatial convolution, and pointwise projection, N_s counts shared experts, and R_2 is channel normalization with learnable scaling. Each iteration executes two LoopLayers, each containing 16 routed and 2 shared experts and activating 4 routed experts. The iteration output $\mathbf{h}_\tau$ proceeds to further refinement or reconstruction.

3.6 Output Reconstruction and Dynamic Depth Scheduling

After K refinement iterations, the decoder D restores $\mathbf{h}_K$ to the input resolution. The residual fusion network F_{fuse} combines it with shallow features $\mathbf{h}_s$ to reconstruct $\hat{\mathbf{Y}}$:

$$\hat{\mathbf{Y}} = \mathrm{clip}_{[0,1]}(\mathbf{X} + F_{\text{fuse}}([D(\mathbf{h}_K); \mathbf{h}_s])), \tag{12}$$

where features are cropped to the original size before fusion. The output preserves the input temporal length and spatial resolution and is clipped to $[0, 1]$.

To adapt computation to the input, the depth predictor maps $[\mathbf{z}; s]$ to probabilities p_4, p_5 of requiring at least four or five iterations, progressively determining K:

$$K = 3 + \mathbb{I}[p_4 \geq 0.5] + \mathbb{I}[p_4 \geq 0.5 \wedge p_5 \geq 0.5], \tag{13}$$

where $\mathbb{I}$ is the indicator function, and p_4, p_5 are Sigmoid outputs of predicted logits. This selects $K \in \{3, 4, 5\}$, permitting five iterations only when the four-iteration condition holds.

3.7 Loss Functions and Depth Supervision

Training proceeds in two stages. The first stage freezes the depth predictor and optimizes restoration at four iterations; the second freezes the restoration network and condition extractor to train the depth predictor. The stage-wise total loss is

$$\mathcal{L}_{\text{total}} = \begin{cases} \mathcal{L}_{\text{rec}}^{(4)} + \mathcal{L}_{\text{aux}}^{(4)}, & \text{Stage 1,} \\ \mathcal{L}_K, & \text{Stage 2.} \end{cases} \tag{14}$$

The restoration loss combines pixel, temporal-difference, and frequency terms; the auxiliary loss constrains modulation and expert routing. The ordinal loss $\mathcal{L}_K$ supervises depth prediction. Appendix A.2 defines these terms and the quality-cost criterion used to construct depth labels.

4 Experiments

4.1 Experimental Setup

Data and evaluation. We use the official UHV-4K-AIO split (Wang et al., 2026b) and one model for all four restoration tasks. Five-frame windows are evaluated on their center frames at native 3840×2160 resolution. We report restoration quality and temporal consistency metrics with equal weighting across tasks; dataset splits, metric definitions, and evaluation protocols are detailed in Appendix B.1–B.3.

Implementation and comparisons. Training uses AdamW (Loshchilov & Hutter, 2019) and FP16 mixed precision (Micikevicius et al., 2018) on one NVIDIA V100 (32GB), with aligned 256×256 crops, a batch size of 2, gradient clipping, and cosine learning rate decay with warmup. The restoration network is first trained for 100 epochs with four iterations. The restoration network and condition extraction module are then frozen, depth labels are generated as detailed in Appendix A.2, and the depth predictor is trained for another 100 epochs. We compare methods on the official UHV-4K-AIO test split using the same evaluation metrics.

4.2 Main Results on UHV-4K-AIO

Table 1 shows that LoopMoEVR attains the highest mean PSNR with 0.884M parameters, exceeding MoCRA by 0.64 dB with roughly one-quarter as many parameters. A single configuration covers all four degradations and achieves the best PSNR and SSIM for dehazing and low-light enhancement. Figure 4 compares identically cropped details from six video methods.

Method	Venue	Dehaze			Derain			Denoise			Low light			Mean			No-reference		Efficiency at 4K	
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	MUSIQ↑	MANIQA↑	Params. (M)	Reported time (s)
Degraded input	–	13.19	0.769	0.226	27.24	0.916	0.131	27.73	0.558	0.493	9.06	0.232	0.966	19.31	0.619	0.454	31.14	0.223	–	–
(a) Image methods																				
AirNet	CVPR'22	22.04	0.939	0.051	28.99	0.973	0.022	37.54	0.968	0.036	19.86	0.826	0.221	27.11	0.927	0.083	36.04	0.230	8.93	46.69
TransWeather	CVPR'22	19.39	0.883	0.105	27.94	0.918	0.129	35.15	0.940	0.110	20.98	0.765	0.347	25.87	0.877	0.173	33.50	0.222	38.05	1.72
UHD-processer	CVPR'25	20.38	0.899	0.106	29.79	0.934	0.113	33.09	0.784	0.340	23.82	0.693	0.490	26.77	0.828	0.262	32.60	0.210	1.71	4.07
DRNet	TMM'26	14.16	0.812	0.183	26.34	0.916	0.127	24.29	0.806	0.270	9.67	0.369	0.686	18.62	0.726	0.317	33.06	0.212	16.49	18.93
R2R	CVPR'26	11.74	0.752	0.264	16.49	0.854	0.163	25.68	0.880	0.177	9.08	0.332	0.676	15.75	0.705	0.320	31.34	0.219	27.58	1.74
(b) Video methods																				
EDVR	CVPRW'19	21.08	0.921	0.093	23.40	0.937	0.081	24.11	0.636	0.272	19.25	0.533	0.412	21.96	0.757	0.214	34.12	0.212	2.85	21.19
BasicVSR++	CVPR'22	21.78	0.932	0.061	26.31	0.955	0.059	34.55	0.948	0.062	22.34	0.819	0.246	26.25	0.914	0.107	35.81	0.218	6.45	43.93
RVRT	NeurIPS'22	19.77	0.827	0.191	23.63	0.899	0.148	19.83	0.842	0.189	17.29	0.671	0.449	20.13	0.810	0.244	35.79	0.212	3.06	76.62
Shift-Net	CVPR'23	22.24	0.924	0.054	27.28	0.919	0.128	36.19	0.917	0.142	21.76	0.765	0.352	26.87	0.881	0.169	34.22	0.219	2.17	9.22
ViWS-Net	ICCV'23	22.23	0.913	0.068	23.59	0.907	0.096	28.57	0.894	0.154	8.44	0.279	0.573	20.71	0.748	0.223	33.08	0.221	57.67	8.91
AverNet	NeurIPS'24	21.25	0.922	0.063	27.58	**0.957**	**0.039**	**37.32**	**0.957**	**0.050**	21.54	0.830	**0.222**	26.92	0.917	**0.094**	**36.07**	0.225	13.72	51.63
MoCRA	arxiv'2026	23.47	0.932	0.050	**28.70**	0.931	0.095	32.07	0.907	0.165	24.66	0.834	0.252	27.23	0.901	0.141	34.40	0.224	3.64	**0.48**
LoopMoEVR (Ours)	-	**24.12**	**0.941**	**0.046**	28.20	0.948	0.081	33.85	0.933	0.138	**25.30**	**0.849**	0.231	**27.87**	**0.918**	0.124	35.62	**0.226**	**0.884**	0.51

Table 1: UHV-4K-AIO restoration results at native 4K.

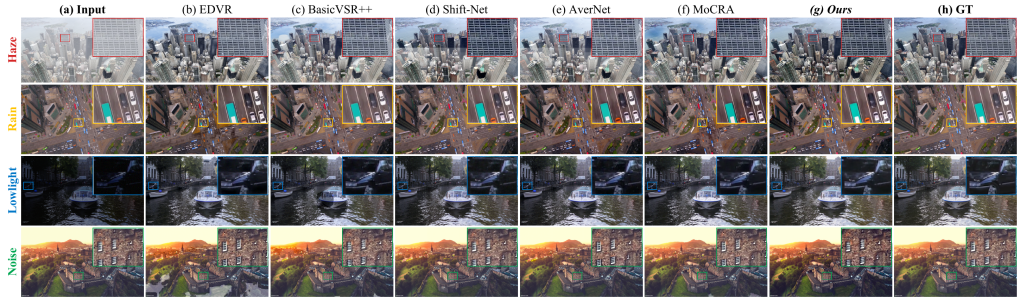

Figure 4: Qualitative comparison of the top six video methods by mean PSNR on UHV-4K-AIO.

4.3 Robustness to Time-Varying Degradations

To evaluate robustness to within-clip degradation changes, we switch degradation types every $\Delta t \in \{6, 3\}$ frames in UHV-4K-AIO test clips. LoopMoEVR achieves 27.68 and 27.85 dB mean PSNR, respectively, at most 0.19 dB below the static setting, indicating robustness to within-clip degradation changes; complete results appear in Appendix B.4.

4.4 Temporal Stability

LoopMoEVR obtains a tLPIPS of 0.1182 and a Warping Error of 0.0297, compared with AverNet's 0.1238 and 0.0296, respectively. The full comparison appears in Appendix B.4. These results demonstrate that the compact model maintains competitive inter-frame consistency at both levels.

Temporal difference supervision constrains inter-frame changes during training, while video-state modulation incorporates within-clip spatio-temporal information during restoration. These mechanisms address temporal consistency through the optimization objective and feature updates, respectively. Together with the main results, the evaluation shows that LoopMoEVR improves average restoration quality while maintaining competitive temporal stability.

4.5 Analysis of Non-UHD Resolution Results

LoopMoEVR remains effective at non-UHD resolutions but shows a smaller advantage than in UHD restoration (Appendix B.5). Designed for UHD inputs, its looped refinement exploits rich structural and degradation cues; fewer high-frequency details at lower resolutions limit iterative gains.

4.6 Real-World UHD Image Restoration

We apply LoopMoEVR trained on UHV-4K-AIO directly to real-world UHD images. As shown in Appendix D.4, it suppresses real-world degradations while preserving clear local structures and texture details, achieving the best visual restoration quality among the compared methods.

4.7 Ablation Studies

Core components. The full model exceeds the LoopMoE-style video baseline (Chen et al., 2026) by 1.20 dB PSNR (Table 2(a)). All listed components improve restoration; momentum also benefits temporal consistency.

Loop-depth allocation. Dynamic depth saves 8.9% inference time over fixed four-step inference at similar quality and gains 0.51 dB PSNR over distribution-matched random allocation (Table 2(b)), demonstrating effective input-dependent computation.

Loop-conditioning formulation. Low-rank interaction exceeds the concatenation MLP by 0.38 dB PSNR and achieves the lowest tLPIPS (Table 2(c)), supporting the interaction of degradation information and iteration position. Visual comparisons appear in Appendix D.2.

Loss functions. The full objective yields the lowest LPIPS and tLPIPS (Table 2(d)); temporal supervision improves consistency, while frequency and auxiliary terms aid restoration. Figure 5 presents visual comparisons.

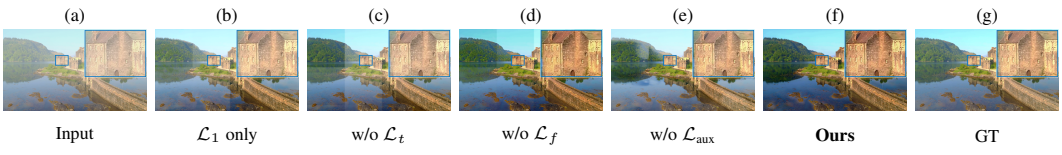

Figure 5: Visual comparison of loss-function ablations.

(a) Core components

Configuration	PSNR	tLPIPS
LoopMoE-style baseline	26.78	0.1614
w/o degr. residual	27.12	0.1424
w/o dual-branch cond.	27.00	0.1484
w/o expert cond.	27.37	0.1354
w/o momentum state	27.30	0.1434
Full model (Ours)	**27.98**	**0.1144**

(b) Loop-depth allocation

Policy	Mean K	PSNR	tLPIPS	ms
Fixed $K = 2$	2.00	26.35	0.1694	**383**
Fixed $K = 3$	3.00	27.24	0.1394	470
Fixed $K = 4$	4.00	27.98	**0.1144**	560
Fixed $K = 5$	5.00	28.08	0.1154	650
Fixed $K = 6$	6.00	**28.11**	0.1169	748
Matched random	3.45	27.36	0.1422	532
Dynamic (Ours)	3.45	27.87	0.1182	510

(c) Loop conditioning

Form	Params (k)	PSNR	tLPIPS
Base iteration encoding	8.0	27.12	0.1424
Additive conditioning	14.0	27.41	0.1324
Concatenation MLP	14.0	27.60	0.1254
Low-rank (Ours)	14.0	**27.98**	**0.1144**

(d) Loss functions

Configuration	PSNR	LPIPS	tLPIPS
$\mathcal{L}_1$ only	26.96	0.159	0.1684
w/o $\mathcal{L}_t$	**28.06**	0.127	0.1524
w/o $\mathcal{L}_f$	27.49	0.145	0.1234
w/o $\mathcal{L}_{\text{aux}}$	27.40	0.143	0.1364
Full loss (Ours)	27.98	**0.122**	**0.1144**

Table 2: LoopMoEVR ablations. Panels (a), (c), and (d) use four-iteration models; panel (b) uses one checkpoint trained at four iterations.

5 Discussion: Why Does Looped Computation Benefit Unified UHD Video Restoration?

Unified 4K restoration must address haze and low light visible at reduced resolution, alongside rain and noise requiring native-scale information (Wang et al., 2026b; Liu et al., 2025). Backbone reuse increases processing depth with shared parameters (Dehghani et al., 2019):

$$\mathbf{h}_\tau = F_\theta(\mathbf{h}_{\tau-1}; \mathbf{e}_\tau, \mathbf{m}_\tau), \qquad P_{\text{shared}} \approx P_\theta, \quad P_{\text{independent}} \approx KP_\theta,$$

Here F_θ updates video features conditioned on loop embedding $\mathbf{e}_\tau$ and momentum state $\mathbf{m}_\tau$. Its P_θ backbone parameters are reused over K iterations, whereas independent backbones require approximately KP_θ parameters. LoopMoEVR refines compact spatio-temporal features and fuses native-resolution shallow features for 4K reconstruction.



Figure 6: Iterative dehazing results of LoopMoEVR.

Degradation, video-state, and iteration conditions adapt shared-backbone updates to the restoration state; prior work supports the role of timestep encoding (Xu & Sato, 2025). Figure 6 illustrates progressive correction, and Appendix D.3 shows that the normal condition order outperforms reversed, shuffled, and repeated alternatives. The parameter-sharing comparison appears in Appendix C.7. LoopMoEVR combines state-dependent refinement with native-resolution reconstruction to enable unified 4K restoration with a compact model.

6 Conclusion

We propose LoopMoEVR, a unified UHD video restoration model based on the loop learning paradigm. It combines degradation-conditioned low-rank loop embeddings, IterAda3DN, and state-aware expert modulation to enable unified UHD video restoration with adaptive depth. With approximately 0.884M parameters, the model achieves the highest mean PSNR on UHV-4K-AIO while maintaining strong performance under time-varying degradations and in temporal-consistency evaluation. The mobile demonstration indicates its potential for resource-constrained deployment.

AI Use Statement

Generative AI tools were used under the authors' direction to assist with the development and refinement of hypotheses, experimental design, software implementation, literature search, result interpretation, and the drafting and editing of the manuscript. They were also used to assist with mathematical derivations and proof writing. All AI-assisted code, analyses, proofs, figures, and text were reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

References

Sangmin Bae, Yujin Kim, Reza Bayat, Sungnyun Kim, Jiyoun Ha, Tal Schuster, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Aaron Courville, and Se-Young Yun. Mixture-of-Recursions: Learning Dynamic Recursive Depths for Adaptive Token-Level Computation. In *NeurIPS*, 2025.

Wenkai Chen, Tianshu Li, Wenyong Huang, Yichun Yin, Lifeng Shang, and Chengwei Qin. Loop-MoE: Unifying Iterative Computation with Mixture-of-Experts for Language Modeling. arXiv preprint arXiv:2606.04438, 2026. URL https://arxiv.org/abs/2606.04438.

Yu-Wei Chen and Soo-Chang Pei. Always Clear Days: Degradation Type and Severity Aware All-in-One Adverse Weather Removal. *IEEE Access*, 13:7650–7662, 2025.

Yuanshuo Cheng, Mingwen Shao, Yecong Wan, Yuanjian Qiao, Wangmeng Zuo, and Deyu Meng. Cross-Consistent Deep Unfolding Network for Adaptive All-In-One Video Restoration. arXiv preprint arXiv:2309.01627, 2023. URL https://arxiv.org/abs/2309.01627.

Zihan Cheng, Liangtai Zhou, Dian Chen, Ni Tang, Xiaotong Luo, Yuan Xie, and Yanyun Qu. UniLDiff: Unlocking the Power of Diffusion Priors for All-in-One Image Restoration. In *CVPR*, pp. 37465–37475, 2026.

Yuning Cui, Syed Waqas Zamir, Salman Khan, Alois Knoll, Mubarak Shah, and Fahad Khan. AdaIR: Adaptive All-in-One Image Restoration via Frequency Mining and Modulation. In *ICLR*, pp. 101306–101327, 2025.

Damai Dai, Chengqi Deng, Chenggang Zhao, R.X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y.K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture-of-Experts Language Models. In *ACL*, pp. 1280–1297, 2024.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal Transformers. In *ICLR*, 2019.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up Test-Time Compute with Latent Reasoning: A Recurrent Depth Approach. arXiv preprint arXiv:2502.05171, 2025. URL https://arxiv.org/abs/2502.05171.

Sahil Goyal, Swayam Agrawal, Gautham Govind Anil, Prateek Jain, Sujoy Paul, and Aditya Kusupati. ELT: Elastic Looped Transformers for Visual Generation. In *ECCV*, pp. 308–326, 2026.

Muhammad Kamran Janjua, Amirhosein Ghasemabadi, Kunlin Zhang, Mohammad Salameh, Chao Gao, and Di Niu. Grounding Degradations in Natural Language for All-In-One Video Restoration. In *WACV*, pp. 5734–5743, 2026.

Ahmadreza Jeddi, Marco Ciccone, and Babak Taati. LoopFormer: Elastic-Depth Looped Transformers for Latent Reasoning via Shortcut Modulation. In *ICLR*, 2026.

Jin-Hwa Kim, Kyoung-Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. Hadamard Product for Low-rank Bilinear Pooling. In *ICLR*, 2017.

Boyun Li, Xiao Liu, Peng Hu, Zhongqin Wu, Jiancheng Lv, and Xi Peng. All-in-One Image Restoration for Unknown Corruption. In *CVPR*, pp. 17452–17462, 2022.

Wenrui Li, Hongtao Chen, Yao Xiao, Wangmeng Zuo, Jiantao Zhou, Yonghong Tian, and Xiaopeng Fan. All-in-One Video Restoration under Smoothly Evolving Unknown Weather Degradations. arXiv preprint arXiv:2601.00533, 2026. URL https://arxiv.org/abs/2601.00533.

Jingbo Lin, Zhilu Zhang, Wenbo Li, Renjing Pei, Hang Xu, Hongzhi Zhang, and Wangmeng Zuo. UniRestorer: Universal Image Restoration via Adaptively Estimating Image Degradation at Proper Granularity. In *ICLR*, 2026.

Yidi Liu, Dong Li, Xueyang Fu, Xin Lu, Jie Huang, and Zheng-Jun Zha. UHD-processer: Unified UHD Image Restoration with Progressive Frequency Learning and Degradation-aware Prompts. In *CVPR*, pp. 23121–23130, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *ICLR*, 2019.

Hongyuan Adam Lu, Z. L. Victor Wei, Qun Zhang, Jinrui Zeng, Bowen Cao, Lingwei Meng, Mocheng Li, Zezhong Wang, Haonan Yin, Naifu Xue, Minyu Chen, Cenyuan Zhang, Zefan Zhang, Hao Wei, Jiawei Zhou, Haoran Xu, Hao Yang, Ronglai Zuo, Tongda Xu, Yonghao Li, Jian Chen, Hebin Wang, Zeyu Gao, Yang Li, Wei Zhao, Qimin Zhong, Siqi Liu, Yumeng Zhang, Leyan Cui, Zhangyu Wang, and Wai Lam. Looped World Models. arXiv preprint arXiv:2606.18208, 2026. URL https://arxiv.org/abs/2606.18208.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed Precision Training. In *ICLR*, 2018.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual Reasoning with a General Conditioning Layer. In *AAAI*, 2018.

Vaishnav Potlapalli, Syed Waqas Zamir, Salman Khan, and Fahad Shahbaz Khan. PromptIR: Prompting for All-in-One Image Restoration. In *NeurIPS*, pp. 71275–71293, 2023.

Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J. Reddi. Reasoning with Latent Thoughts: On the Power of Looped Transformers. In *ICLR*, 2025.

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. In *ICLR*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *NeurIPS*, 2017.

Ye-Cong Wan, Ming-Wen Shao, Yuan-Shuo Cheng, Jun Shu, and Shui-Gen Wang. Removing Multiple Hybrid Adverse Weather in Video via a Unified Model. *IEEE TCSVT*, 36(7):9687–9700, 2026.

Shaowen Wang, Ge Zhang, Kairong Luo, Yuhao Wu, Shaofan Liu, Jiaheng Liu, Wenhao Huang, Shen Yan, and Jian Li. SMELT: Scaling Laws for Compute-Matched MoE Looped Transformers. arXiv preprint arXiv:2609.01343, 2026a. URL https://arxiv.org/abs/2609.01343.

Yongcong Wang, Pu Wang, Hingchin Chen, Runci Bai, Yucheng Xin, Chen Wu, Chengchao Shen, Guangwei Gao, Siyuan Yao, Pengwen Dai, and Zhuoran Zheng. MoCRA: Mixture of Compositional Rank-1 Atoms for 4K All-in-One Video Restoration. arXiv preprint arXiv:2608.01829, 2026b. URL https://arxiv.org/abs/2608.01829.

Kevin Xu and Issei Sato. On Expressive Power of Looped Transformers: Theoretical Analysis and Enhancement via Timestep Encoding. In *ICML*, volume 267, pp. 69613–69646, 2025.

Eduard Zamfir, Zongwei Wu, Nancy Mehta, Yuedong Tan, Danda Pani Paudel, Yulun Zhang, and Radu Timofte. Complexity Experts are Task-Discriminative Learners for Any Image Restoration. In *CVPR*, pp. 12753–12763, 2025.

Biao Zhang and Rico Sennrich. Root Mean Square Layer Normalization. In *NeurIPS*, 2019.

Haiyu Zhao, Lei Tian, Xinyan Xiao, Peng Hu, Yuanbiao Gou, and Xi Peng. AverNet: All-in-one Video Restoration for Time-varying Unknown Degradations. In *NeurIPS*, 2024.

Haiyu Zhao, Yiwen Shan, Yuanbiao Gou, and Xi Peng. LaverNet: Lightweight All-in-one Video Restoration via Selective Propagation. arXiv preprint arXiv:2512.16313, 2025. URL https://arxiv.org/abs/2512.16313.

Appendix Contents

The supplementary material is organized into four parts: Section A specifies the model and training objective; Section B documents the evaluation protocol; Section C reports quantitative ablations; and Section D discusses supplementary visual evidence.

A. Architecture and Training Details 13

B. Experimental Protocol and Reproducibility 14

C. Extended Ablations 16

D. Qualitative Evidence and Mechanism Analysis 18

A Architecture and Training Details

A.1 Forward Pass and Module Interfaces

For each five-frame input window, the encoder produces spatially downsampled latent features and the shallow branch retains full-resolution detail. Global pooling of the latent features and input channel statistics provides the degradation representation and severity-related scalar. Their low-rank interaction with the iteration encoding yields the stage condition. At each iteration, clip-level momentum and the current features determine a modulation mask; two parameter-shared LoopLayers then update the spatio-temporal representation. Each LoopLayer has 16 routed and 2 shared experts, with 4 routed experts activated for a clip. The attention branch precedes the expert branch, so expert modulation uses the attention-updated local state. The decoder fuses the final latent state with shallow features and predicts a residual at the input resolution. The iteration count is selected from $\{3, 4, 5\}$ by the depth predictor.

A.2 Two-Stage Training Objective

Training separates restoration learning from depth prediction. Given the predicted video $\hat{\mathbf{Y}}$ and reference video $\mathbf{Y}$, we first construct a restoration loss with pixel, temporal, and frequency constraints:

$$\begin{aligned}\mathcal{L}_{\text{rec}} = &\|\hat{\mathbf{Y}} - \mathbf{Y}\|_1 + \lambda_t \|\Delta_t \hat{\mathbf{Y}} - \Delta_t \mathbf{Y}\|_1 \\ &+ \lambda_f \left\| |\mathcal{F}_{HW}(\hat{\mathbf{Y}})| - |\mathcal{F}_{HW}(\mathbf{Y})| \right\|_1,\end{aligned} \tag{A.1}$$

where Δ_t denotes inter-frame differences, $\mathcal{F}_{HW}$ is the spatial Fourier transform, and λ_t, λ_f are the corresponding weights.

An auxiliary loss further constrains loop modulation and expert routing:

$$\mathcal{L}_{\text{aux}} = \lambda_m(\mathcal{L}_{\text{mask}} + \mathcal{R}_{\text{overlap}}) + \lambda_b \mathcal{R}_{\text{load}} + \lambda_c \mathcal{L}_{\text{cls}}, \tag{A.2}$$

where the terms constrain adjacent-iteration masks, expert selection overlap, load balancing, and degradation classification, respectively, with weights $\lambda_m, \lambda_b, \lambda_c$.

The first stage freezes the depth predictor and optimizes the restoration and auxiliary terms in Eq. (14) for 100 epochs at four iterations. The superscript (4) denotes losses computed from four-iteration outputs and intermediate states, with cross-iteration momentum updated according to Eq. (4).

The second stage freezes the restoration network and condition extractor. Candidate depths are evaluated using the quality-cost criterion

$$\mathcal{J}(K) = \mathcal{L}_{\text{rec}}^{(K)} + \eta_t \widetilde{T}(K) + \eta_m \widetilde{M}(K), \tag{A.3}$$

where $\widetilde{T}$ and $\widetilde{M}$ are normalized inference time and GPU memory costs, respectively, weighted by η_t and η_m.

Using $\mathcal{J}(K)$, we assign each training window a target depth K^*:

$$K^* = \min\left(\arg\min_{K \in \{3,4,5\}} \mathcal{J}(K) \right), \tag{A.4}$$

where K^* minimizes the objective, with ties resolved in favor of the smaller K.

To learn this selection, K^* is converted into ordinal labels y_4, y_5, supervising predicted logits a_4, a_5:

$$y_j = \mathbb{I}[K^* \geq j], \qquad \mathcal{L}_K = \frac{1}{2} \sum_{j \in \{4,5\}} \text{BCE}(\sigma(a_j), y_j), \tag{A.5}$$

where y_j indicates whether at least j iterations are required, and BCE denotes binary cross-entropy. Only the depth predictor is updated for 100 epochs in this stage:

$$\mathcal{L}_{\text{stage2}} = \mathcal{L}_K. \tag{A.6}$$

This training conditions depth selection on input degradation. At inference, the iteration count is predicted directly without reference videos or reevaluation of the quality-cost objective.

A.3 Inference-Time Depth Selection

The restoration network and condition extractor remain fixed after the first stage. During predictor training, each window is assigned the lowest-cost depth among $3, 4, 5$ according to Eq. (A.3); ties favor the smaller depth. During inference, two predicted threshold probabilities determine whether the fourth and fifth iterations are executed, as in the main text. Reference frames, quality-cost evaluation, and per-depth reruns are not required at inference.

B Experimental Protocol and Reproducibility

B.1 Data and Preprocessing

Experiments use the UHV-4K-AIO official split of 80 training and 20 test videos across dehazing, deraining, denoising, and low-light enhancement. A single model is used across the four tasks. Training samples aligned 256×256 crops; testing processes five-frame windows and evaluates their center frames at 3840×2160 resolution. All reported task means give equal weight to the four degradation categories.

B.2 Training Details

Both training stages run for 100 epochs. Training uses AdamW (Loshchilov & Hutter, 2019), FP16 mixed precision (Micikevicius et al., 2018), a batch size of 2, gradient clipping, and cosine learning-rate decay with warmup on one NVIDIA V100 (32GB). The restoration network is optimized with four iterations in the first stage; the condition extractor and restoration network are frozen when training the depth predictor in the second stage.

B.3 Metrics and Temporal Evaluation

Spatial reconstruction is reported with PSNR and SSIM; perceptual quality is reported with LPIPS, MUSIQ, and MANIQA. Temporal consistency is evaluated using dataset-provided reference optical flow for Warping Error and LPIPS features for tLPIPS. Lower Warping Error and tLPIPS indicate better temporal consistency. The complete temporal comparison appears in Appendix B.4.

B.4 Time-Varying Degradations and Temporal Consistency

Table 3 reports restoration quality when the degradation type switches every $\Delta t \in \{6, 3\}$ frames. Table 4 compares temporal stability averaged across the four tasks; LoopMoEVR uses its predicted iteration depth.

Method	Venue	Static	$\Delta t = 6$		$\Delta t = 3$	
		PSNR↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
(a) Image methods						
AirNet	CVPR'22	27.11	26.09	0.914	25.76	0.915
TransWeather	CVPR'22	25.87	23.41	0.834	23.27	0.832
DRNet	TMM'26	18.62	23.17	0.820	23.05	0.824
UHD-processer	CVPR'25	26.77	24.05	0.746	24.19	0.752
R2R	CVPR'26	15.75	26.02	0.829	25.59	0.828
(b) Video methods						
EDVR	CVPRW'19	21.96	20.15	0.674	24.95	0.727
BasicVSR++	CVPR'22	26.25	23.87	0.864	25.81	0.885
Shift-Net	CVPR'23	26.87	25.62	0.799	27.14	0.821
RVRT	NeurIPS'22	20.13	19.49	0.790	21.86	0.809
ViWS-Net	ICCV'23	20.71	21.52	0.812	22.28	0.832
AverNet	NeurIPS'24	26.92	22.36	0.714	24.94	0.756
MoCRA	arxiv'2026	27.23	26.89	0.898	27.16	0.901
LoopMoEVR (Ours)	–	**27.87**	**27.68**	**0.916**	**27.85**	**0.915**

Table 3: Average restoration performance under time-varying degradations.

Method	Venue	Warping Error↓	tLPIPS↓
AirNet	CVPR'22	0.0305	0.1256
TransWeather	CVPR'22	0.0329	0.1420
UHD-processer	CVPR'25	0.0327	0.1551
AverNet	NeurIPS'24	**0.0296**	0.1238
MoCRA	arxiv'2026	0.0300	0.1375
LoopMoEVR (Ours)	–	0.0297	**0.1182**

Table 4: Temporal stability averaged across four tasks. Warping Error is computed using the same reference optical flow for all methods.

B.5 Non-UHD Resolution Results

Table 5 reports PSNR and SSIM on MOT17 and DAVIS-test under two evolving-degradation settings (Li et al., 2026). Setting 1 varies degradation severity within each video, while Setting 2 also changes the degradation type. LoopMoEVR and the comparison methods were trained and evaluated under the same respective settings. Average denotes the arithmetic mean of Settings 1 and 2 for each dataset.

Table 5: Restoration results on MOT17 and DAVIS-test under two evolving-degradation settings.

Method	MOT17						DAVIS-test					
	Setting 1		Setting 2		Average		Setting 1		Setting 2		Average	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
TransWeather	18.73	0.5506	19.94	0.6145	19.34	0.5826	20.44	0.5672	20.64	0.5839	20.54	0.5756
AirNet	23.62	0.8900	26.56	0.9412	25.09	0.9156	24.65	0.9034	19.93	0.8774	22.29	0.8904
BasicVSR++	36.26	0.9793	32.43	0.9787	34.35	0.9790	32.33	0.9739	25.51	0.9527	28.92	0.9633
RVRT	25.63	0.8851	23.36	0.9040	24.50	0.8946	27.71	0.9225	19.39	0.8313	23.55	0.8769
AverNet	28.98	0.8894	27.42	0.8582	28.20	0.8738	25.46	0.8023	25.01	0.7952	25.24	0.7988
LoopMoEVR (Ours)	31.95	0.9184	30.89	0.9070	31.42	0.9127	27.61	0.8921	26.75	0.8831	27.18	0.8876

B.6 Anonymous User Study

One hundred participants anonymously rated six methods on a common three-point visual-quality scale (3 = best), considering local details, color and brightness, and temporal consistency. LoopMoEVR received the most top ratings (47/100) and the highest mean opinion score (MOS 2.35), compared with MoCRA (39/100; MOS 2.21). Figure 7 shows the rating distribution and MOS for each method.

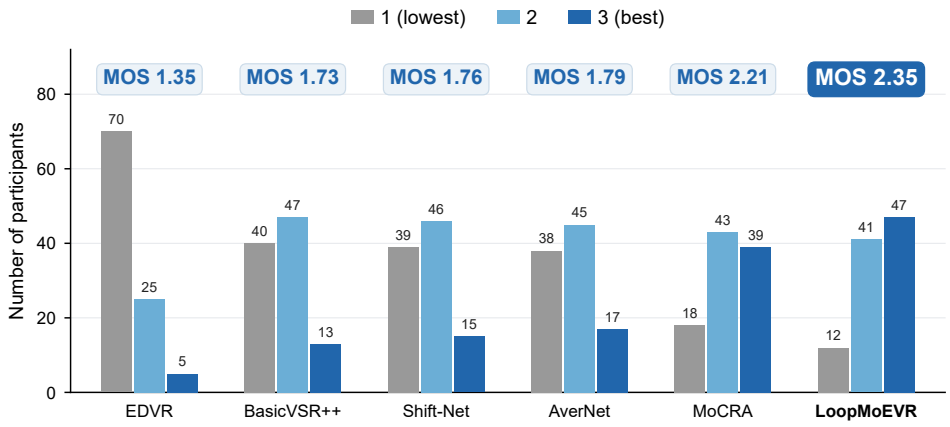

Figure 7: User-study rating distribution (100 participants; 3 = best). Mean opinion scores (MOS) are shown above each method.

C Extended Ablations

C.1 Core Components

Table 2(a) in the main paper summarizes the core-component ablations. Except for depth allocation, variants are independently trained for four iterations with the same data split and training budget. Video-conditioning ablations remove only the corresponding pooled-state and momentum inputs while retaining local-feature and iteration conditioning. A qualitative comparison appears in Figure 8.

C.2 Loop-Conditioning Formulations

Table 2(c) in the main paper compares the base iteration encoding, additive degradation conditioning, a concatenation MLP, and the low-rank interaction. The three degradation-conditioned alternatives have matched conditioning-generator parameter budgets and residual-output initialization. The corresponding visualization appears in Figure 9.

C.3 Expert-Branch Architecture Ablation

To evaluate the expert-branch architecture, we fix the iteration count at $K = 4$ and compare two dense replacement branches, a branch retaining only routed experts, and the full model under the same training and evaluation settings. Table 6 shows that the routed-only variant outperforms both dense alternatives. Adding shared experts further yields the best result among the evaluated configurations, improving PSNR by 1.03 dB over the routed-only variant. Within the tested architectures, the full expert branch achieves stronger overall restoration performance, supporting the effectiveness of jointly using routed and shared experts for unified video restoration.

Expert branch	PSNR↑	LPIPS↓	tLPIPS↓
Compact Dense	25.83	0.162	0.1484
Expanded Dense	26.67	0.153	0.1392
Routed experts only	26.95	0.137	0.1289
Full model (Ours)	**27.98**	**0.122**	**0.1144**

Table 6: Expert-branch architecture ablation at $K = 4$.

C.4 Loop-Depth Allocation

Table 2(b) in the main paper compares fixed, random, and input-adaptive depth policies using the same four-iteration checkpoint. The random and dynamic policies have the same within-task depth distribution. Relative to $K = 3$, $K = 2$ saves 87 ms but loses 0.89 dB PSNR and increases tLPIPS by 0.0300. Relative to $K = 5$, $K = 6$ costs 98 ms for a 0.03 dB PSNR gain and slightly higher tLPIPS. Considering the restoration quality and inference time of the fixed-depth variants, we restrict the adaptive depth candidates to $\{3, 4, 5\}$.

C.5 Loss-Function Components

Table 2(d) in the main paper reports independently trained four-iteration variants with individual loss terms omitted while the remaining weights are held fixed. Removing temporal supervision primarily weakens tLPIPS; removing frequency or auxiliary terms reduces PSNR and perceptual quality. The visual comparison appears in Figure 5 of the main paper.

C.6 Ablation of Looped Temporal Modeling

To assess the effectiveness of our looped temporal modeling, we fix the iteration count at $K = 4$ and modify how state conditions are constructed and used for modulation under the same training settings. We also examine the joint removal of the momentum state and temporal loss $\mathcal{L}_t$. Table 7 shows that the full model achieves the best PSNR, Warping Error, and tLPIPS among the configurations considered. Across the four state-path variants, PSNR decreases by 0.78–0.89 dB and tLPIPS increases by 0.0110–0.0160, supporting the contribution of our looped state modeling to reconstruction accuracy and temporal consistency. Removing both the momentum state and $\mathcal{L}_t$ further reduces PSNR to 26.85 dB and increases tLPIPS to 0.1744, indicating that these two components jointly support temporally stable video restoration.

Configuration	PSNR↑	Warping Error↓	tLPIPS↓
w/o current pooled condition	27.14	0.0312	0.1274
w/o momentum history	27.09	0.0317	0.1304
Modulation with pre-update state	27.20	0.0308	0.1254
w/o loop mask	27.17	0.0310	0.1264
w/o momentum state and $\mathcal{L}_t$	26.85	0.0375	0.1744
Full model (Ours)	**27.98**	**0.0293**	**0.1144**

Table 7: Ablation of looped temporal modeling at $K = 4$.

C.7 Cross-Iteration Parameter Sharing Ablation

To evaluate cross-iteration parameter sharing, we compare shared and independent parameter configurations at matched network width and a fixed depth of four iterations. As shown in Table 8, sharing reduces the parameter count from 3.116M to 0.884M, a reduction of 71.6%, with a PSNR difference of only 0.08 dB and a lower tLPIPS. Through cross-iteration parameter reuse, LoopMoEVR maintains restoration accuracy close to the independent configuration while improving temporal consistency, demonstrating the parameter efficiency of looped computation for unified UHD video restoration.

Parameter configuration	Params/M↓	ΔPSNR/dB↑	tLPIPS↓	Time/ms↓
Unshared across iterations	3.116	**+0.08**	0.1160	**515**
Shared across iterations (Ours)	**0.884**	0.00	**0.1144**	560

Table 8: Cross-iteration parameter sharing ablation. All configurations use four iterations; ΔPSNR is relative to the shared configuration, and time is end-to-end latency per five-frame 4K window.

D Qualitative Evidence and Mechanism Analysis

D.1 Visual Comparisons Retained in the Main Paper

The main paper retains the full six-method and identical-crop comparison across four restoration tasks (Figure 4), the smartphone demonstration (Figure 2), and the iteration-by-iteration dehazing sequence (Figure 6). These displays document complementary aspects of the result: comparative local detail, a deployment example, and progressive correction within a shared-parameter backbone.

D.2 Qualitative Ablation Comparisons

Figure 8 compares the core-component variants, and Figure 9 compares loop-conditioning formulations. The loss-function visualization appears in the main paper (Figure 5).

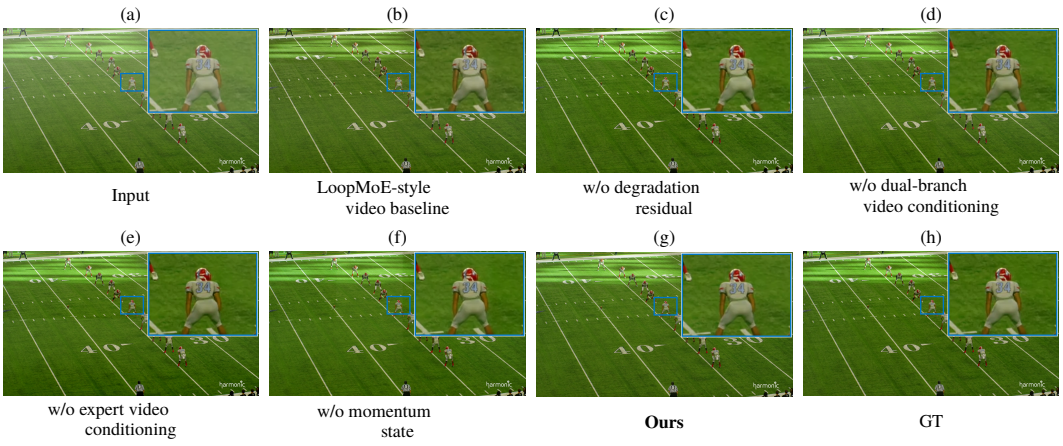

Figure 8: Qualitative comparison of core-component ablations.

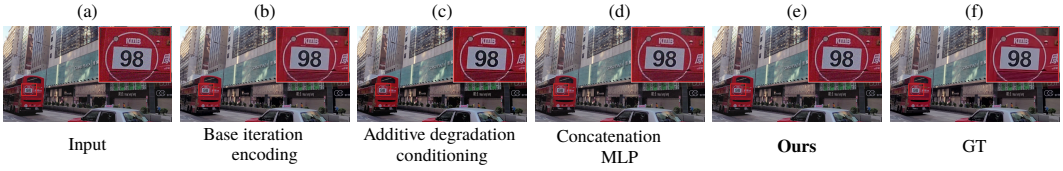

Figure 9: Qualitative comparison of loop-conditioning ablations.

D.3 Loop-Condition Order

Under four iterations, we compare the normal order of loop conditions with reversed, shuffled, and repeated alternatives. Figure 10 shows that the normal order attains higher PSNR and lower tLPIPS than each alternative, supporting the role of stage-specific condition ordering in iterative restoration.

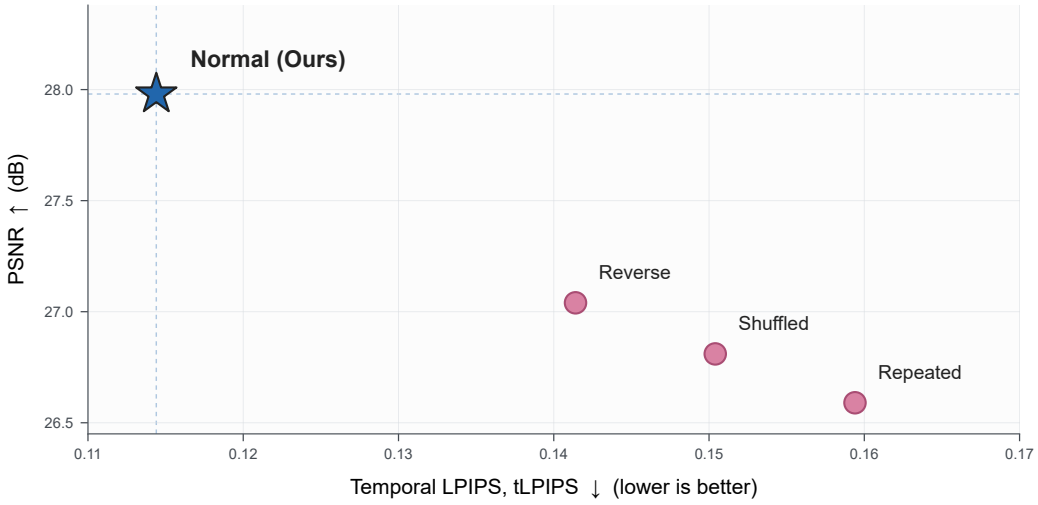

Figure 10: PSNR and tLPIPS under different loop-condition orders at $K = 4$.

D.4 Real-World UHD Image Restoration

Figure 11 presents restoration comparisons on real-world UHD images.

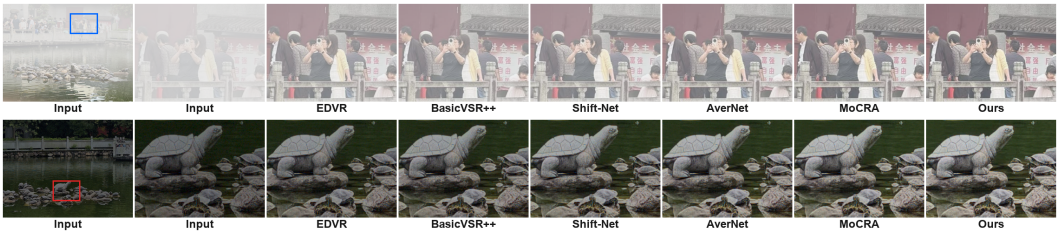

Figure 11: Restoration results on real-world UHD images.