# Learning Phase Distortion with Selective State Space Models for Video Turbulence Mitigation

Supplementary Material

## 1. More details about the architecture

### 1.1. State Space Model

To process the discrete input sequence $\boldsymbol{x} = (x_0, x_1, ..., x_{L-1}) \in \mathbb{R}^L$, following [4], Mamba [2] employs the zero-order hold (ZOH) assumption to convert the continuous parameters $\boldsymbol{A}, \boldsymbol{B}$ into their discrete counterparts $\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}}$ as: $\bar{\boldsymbol{A}} = e^{\boldsymbol{\Delta A}}$, $\bar{\boldsymbol{B}} = (\boldsymbol{\Delta A})^{-1}(e^{\boldsymbol{\Delta A}} - \boldsymbol{I}) \cdot \boldsymbol{\Delta B}$, where $\boldsymbol{\Delta}$ is the time scale. After discretizing $\boldsymbol{A}, \boldsymbol{B}$ to $\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}}$, the SSM can be reformulated as:

$$\boldsymbol{h}_t = \bar{\boldsymbol{A}}\boldsymbol{h}_{t-1} + \bar{\boldsymbol{B}}x_t, \quad y_t = \boldsymbol{C}\boldsymbol{h}_t + \boldsymbol{D}x_t \tag{1}$$

Eq.1 represents a sequence-to-sequence mapping from $x_t$ to $y_t$. Since all operations are linear, all steps can be computed in parallel. To facilitate this, a convolution kernel is constructed [3]: $\mathbf{K} = (\boldsymbol{CB}, \boldsymbol{CAB}, \ldots, \boldsymbol{CA}^{L-1}\boldsymbol{B})$, where the recursive multiplication of $\boldsymbol{A}$ can be efficiently computed by the scan algorithm and final output $\boldsymbol{y}$ is computed by the convolution: $\boldsymbol{y} = \boldsymbol{x} * \mathbf{K}$, which has linear complexity with respect to the length of $\boldsymbol{x}$.

However, $\mathbf{K}$ is static over time, which does not satisfy the requirement of real-world processes. To alleviate this, the selective state space model (S6) [2] models the $\boldsymbol{\Delta}$, $\boldsymbol{B}$, $\boldsymbol{C}$ as linear projections of the input $\boldsymbol{x}$. This operation successfully enables the input-dependent selective property.

### 1.2. The ReBlurNet (RBN)

The RBN initially transforms the input image into multi-scale features, which are then modulated through element-wise multiplication with the multi-scale features of $\widetilde{\mathbf{a}}$ before being decoded to produce the blurred output image. While any U-Net style architecture could serve as the base network for the RBN, we ultimately selected NAFNet for this implementation. Within the RBN framework, the latent blur feature $\mathbf{b}$ undergoes processing through a sequence of encoders, each comprising $1 \times 1$ convolution followed by ReLU activation. The features produced by each encoder are downsampled before being passed to the subsequent encoder. We denote the output features from the four encoders as $\mathbf{eb}^1$, $\mathbf{eb}^2$, $\mathbf{eb}^3$, and $\mathbf{eb}^4$. Concurrently, the input image is processed through the base network to generate the blurred result. Importantly, before each input feature $\mathbf{vi}^i$ enters the $i$-th encoder for processing, it undergoes modulation via element-wise multiplication with $\mathbf{eb}^i$. The decoder component of the base network remains unmodified in our RBN implementation.

| Models | # of params (M) | GMACs | Latency (s) |
|---|---|---|---|
| TSRWGAN [5] | 42.08 | - | 0.85 |
| TMT [9] | 26.04 | 1806.0 | 0.76 |
| DATUM [8] | 5.754 | 372.7 | 0.056 |
| Turb-Seg-Res [7] | $\sim 30$ | - | 2.404 |
| MambaTM [ours] | 6.904 | 143.5 | 0.030 |

Table 1. The cost of different video TM methods. The GMAC and Latency are evaluated framewise under $960 \times 540$ patches with NVIDIA A100 GPUs

| # of input frames | PSNR | SSIM | LPIPS |
|---|---|---|---|
| 30 | 29.5765 | 0.8793 | 0.1544 |
| 40 | 29.6979 | 0.8815 | 0.1530 |
| 60 | 29.8129 | 0.8834 | 0.1521 |
| 120 | 29.9151 | 0.8843 | 0.1516 |

Table 2. The impact of numbers of input frames during inference

## 2. Cost of video TM methods

As an extension of Table 3 in the main paper, we provide the computational cost of MambaTM and other other video TM methods regarding model size and MACs in table 1. Our model requires the least computation cost and has a much faster inference speed than other models.

## 3. Additional experiments

### 3.1. Temporal extrapolation

Same as [6, 8], we can also observe better performance with more input frames during testing. As shown in Table 2, our MambaTM shows good temporal extrapolation properties.

### 3.2. The latent phase distortion (LPD)

We visualize an example of our Zernike VAE and LPD in Figure 2. This example is taken from the validation set, featuring an unseen scene and previously unencountered turbulence parameters. We observe that the re-degraded image produced by LPD and RBN is visually similar to the degraded image generated using the Zernike coefficients. The mean of LPD, $\mu$, represents the turbulence strength, while the variance $\sigma^2$ is visually correlated with the blur strength variation, as indicated by the pixel-wise $L_2$ norm of the corresponding Zernike coefficients.

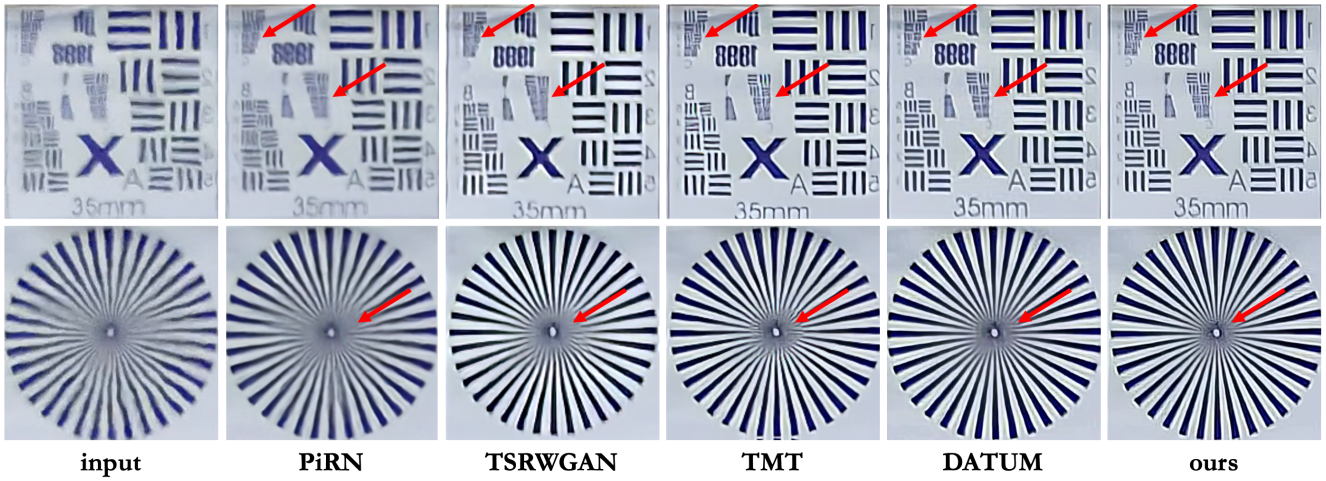



Figure 1. ualitative comparison on the OTIS dataset [1]. The images on the top are from the 13th sequence and the images on the bottom are from the 14th sequence. Zoom in for better view

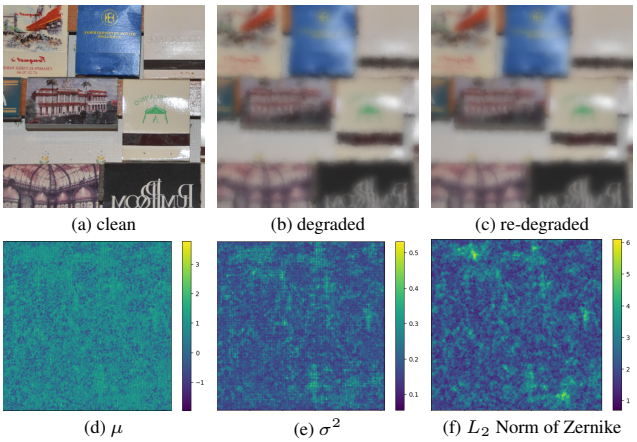



Figure 2. A sample of the Zernike VAE and LPD map. (b) is generated by the Zernike-based simulator with input image (a) and Zernike coefficients whose pixel-wise norm is shown in (f), the blur kernel size is $55 \times 55$. (c) is generated by our RBN with the predicted LPD, whose statistics are shown in (d) and (e). Please zoom in for a better view.

### 3.3. Real-world samples of the LPD-based simulation

To demonstrate the generalization capbility of the LPD estimation and our LPD-based simulator, we provide a real-world testing case in Figure 3. It can be seen that our model successfully recovered the clean patterns from the turbulence-affected images across a long-range distance. By comparing the real-world degraded and our re-degraded images using the restored image as the input, we can find that our simulator can faithfully represent real-world turbulence. We also provide the associated videos in the supplementary material.

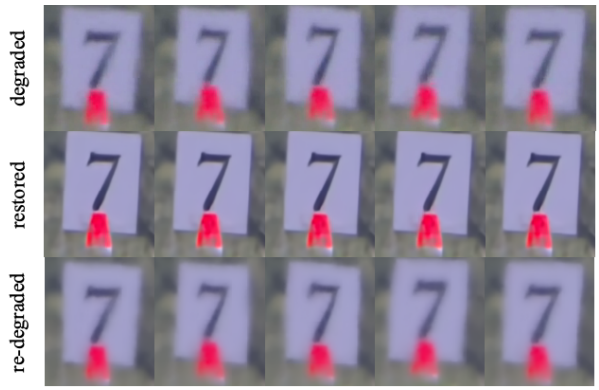
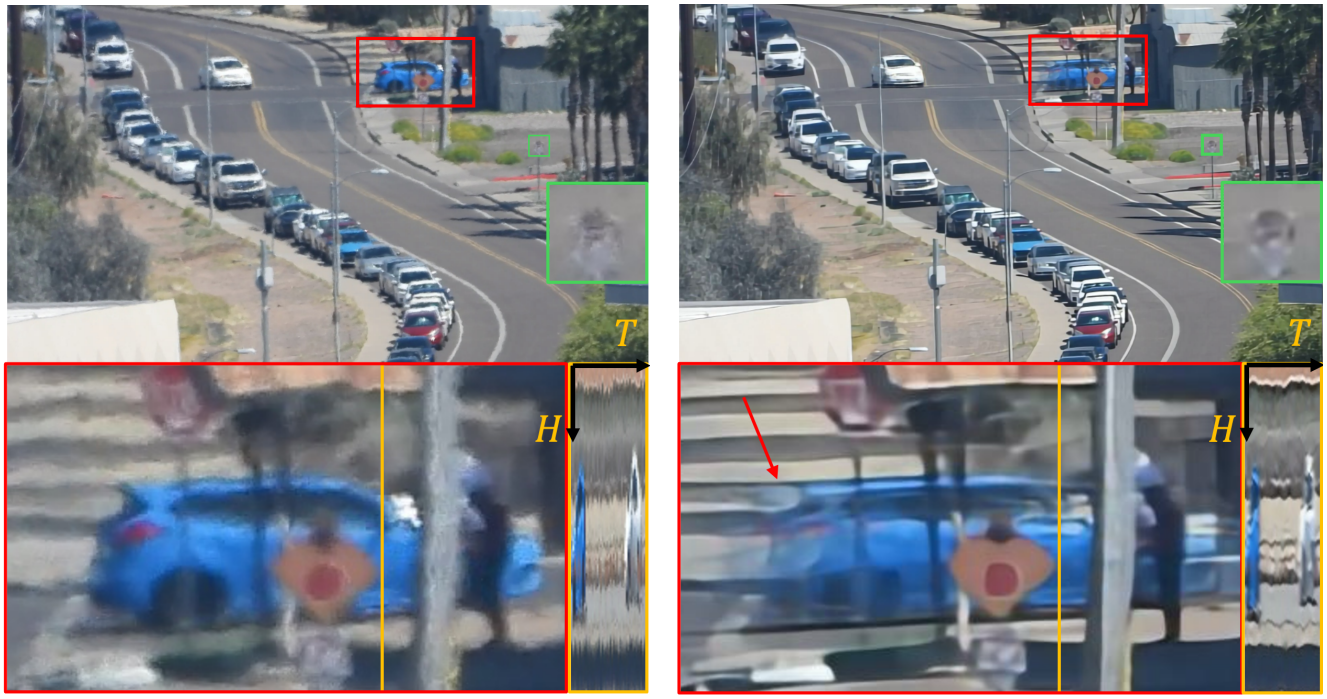



Figure 3. Comparing the real-world turbulent images (from *BRIAR_1* in the supplementary material) and re-degraded images.

### 3.4. More qualitative comparison

To demonstrate the advancement of our method, we further provide two real-world comparisons. The first is on the static scenes from the OTIS dataset [1]. As presented in Figure 1, we compare MambaTM with other SOTA turbulence mitigation works and we can find that our method recovers more details than others. The second is on the dynamic scene from the URG-T dataset [7], we compare MambaTM with two recent SOTA DATUM [8] and Turb-Seg-Res [7]. To highlight our method's temporal consistency on dynamic scenes, we fetch 1D spatial slices from the same location in each frame of the image sequences and stitch all slices along the time axis. The result is shown in Figure 4. From this, we can find that our method shows better restoration quality both spatially and temporally. Meanwhile, notice that our method is $2\times$ faster than DATUM and $50\times$ faster than Turb-Seg-Res.




(a) Input frame




(b) Turb-Seg-Res (CVPR 2024) [7]


(c) DATUM (CVPR 2024) [8]


(d) MambaTM (ours)

Figure 4. Qualitative comparison on the URG-T real-world dataset [7]. From the green box, we can find that *spatially*, our method can produce the sharpest and most reliable restoration. We provide temporal slices (the orange line in red bounding boxes of each frame) in the bottom right of each figure, from which we can find that *temporally*, our method generates the most stable and consistent output. Note Figure (b) also suffers from the ghost effect caused by its temporal fusion method.

## References

[1] Jérôme Gilles and Nicholas B Ferrante. Open turbulent image set (OTIS). *Pattern Recognition Letters*, 86:38 – 41, 2017. 2

[2] Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In *First Conference on Lan-*

*guage Modeling*, 2024. 1

[3] Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022. 1

[4] Ankit Gupta, Albert Gu, and Jonathan Berant. Diagonal state spaces are as effective as structured state spaces. *Advances in Neural Information Processing Systems*, 35:22982–22994, 2022. 1

[5] D. Jin, Y. Chen, Y. Lu, J. Chen, P. Wang, Z. Liu, S. Guo, and X. Bai. Neutralizing the impact of atmospheric turbulence on complex scene imaging via deep learning. *Nature Machine Intelligence*, 3:876 – 884, 2021. 1

[6] Dong Lao, Congli Wang, Alex Wong, and Stefano Soatto. Diffeomorphic template registration for atmospheric turbulence mitigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 25107–25116, 2024. 1

[7] Ripon Kumar Saha, Dehao Qin, Nianyi Li, Jinwei Ye, and Suren Jayasuriya. Turb-Seg-Res: A segment-then-restore pipeline for dynamic videos with atmospheric turbulence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25286–25296, 2024. 1, 2, 3

[8] Xingguang Zhang, Nicholas Chimitt, Yiheng Chi, Zhiyuan Mao, and Stanley H Chan. Spatio-temporal turbulence mitigation: A translational perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2889–2899, 2024. 1, 2, 3

[9] Xingguang Zhang, Zhiyuan Mao, Nicholas Chimitt, and Stanley H. Chan. Imaging through the atmosphere using turbulence mitigation transformer. *IEEE Transactions on Computational Imaging*, 10:115–128, 2024. 1