
Temporally Consistent Atmospheric Turbulence Mitigation with Neural Representations

**Haoming Cai^{*1}, Jingxi Chen^{*1}, Brandon Y. Feng², Weiyun Jiang³, Mingyang Xie¹,
Kevin Zhang¹, Cornelia Fermuller¹, Yiannis Aloimonos¹, Ashok Veeraraghavan³,
Christopher A. Metzler^{†1}**

¹University of Maryland, ²Massachusetts Institute of Technology, ³Rice University

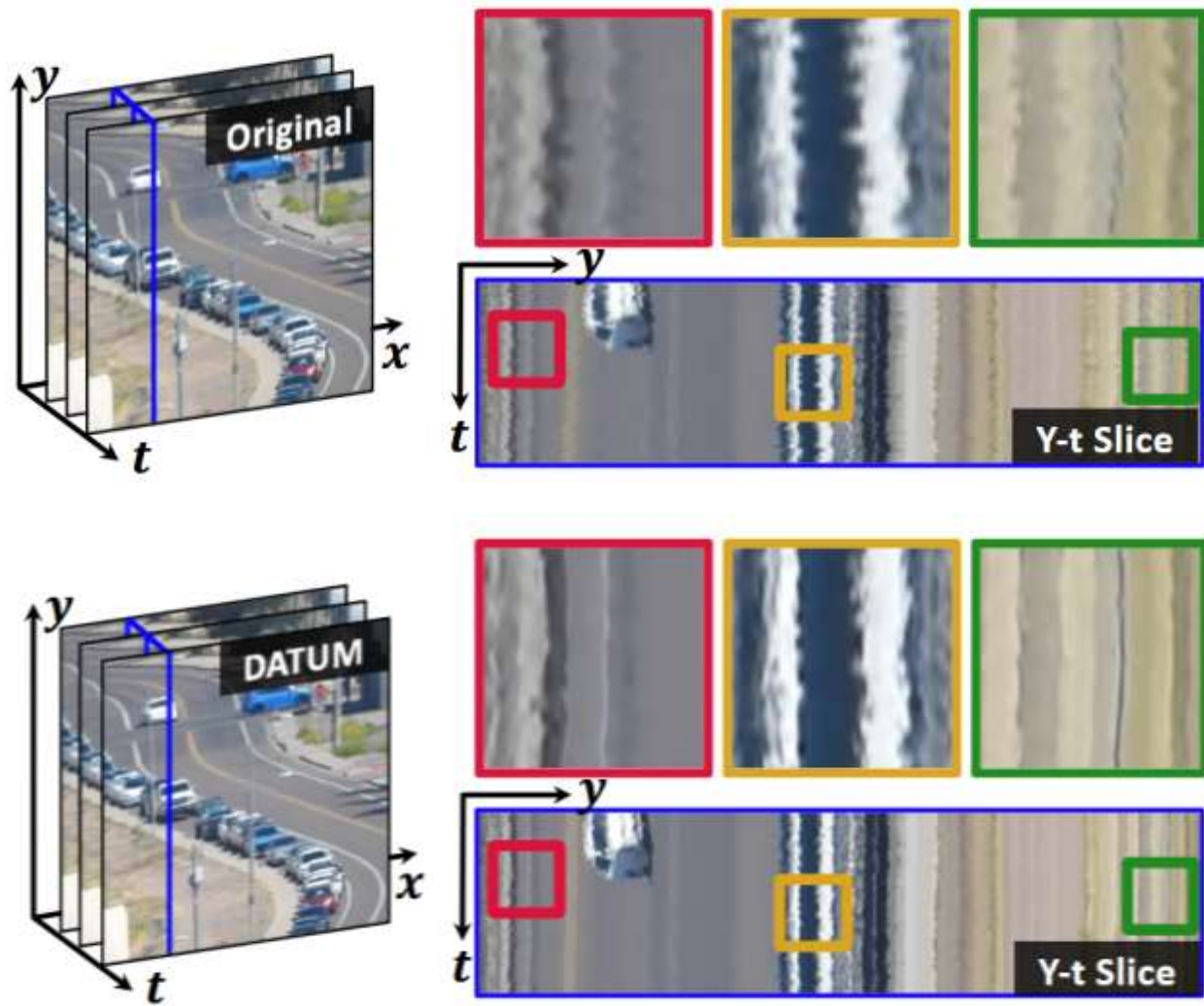
存在问题



Figure 1: **Temporally consistent restoration in video ATM is challenging.** State-of-the-art methods like DATUM (2) (CVPR'24) and TMT (3)(TCI'23), designed for video ATM, fail to maintain temporal consistency in real-world atmospheric turbulence. For instance, they produce flickering artifacts on a static pole.

大气湍流（即空气折射率的随机波动）会在长距离拍摄的图像中引入复杂的**时空扭曲**，导致视频中的图像出现模糊和变形等现象。这种影响不仅是空间上的变化（例如图像**模糊**），还包括时间上的变化，使得不同帧之间的图像出现**不一致的波动**。

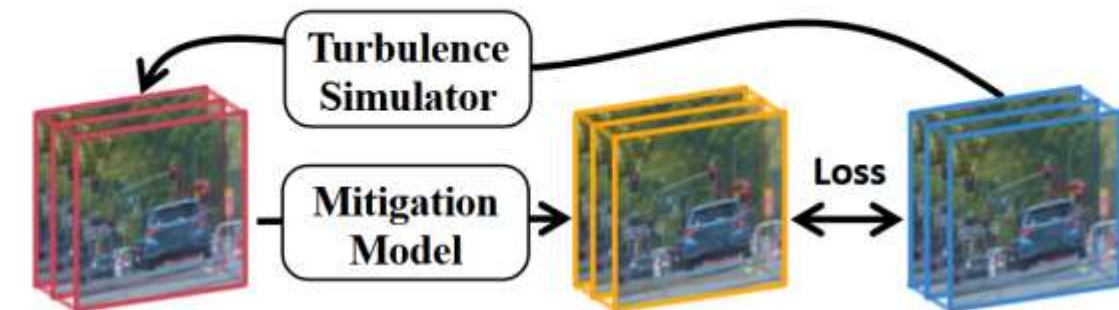
存在问题



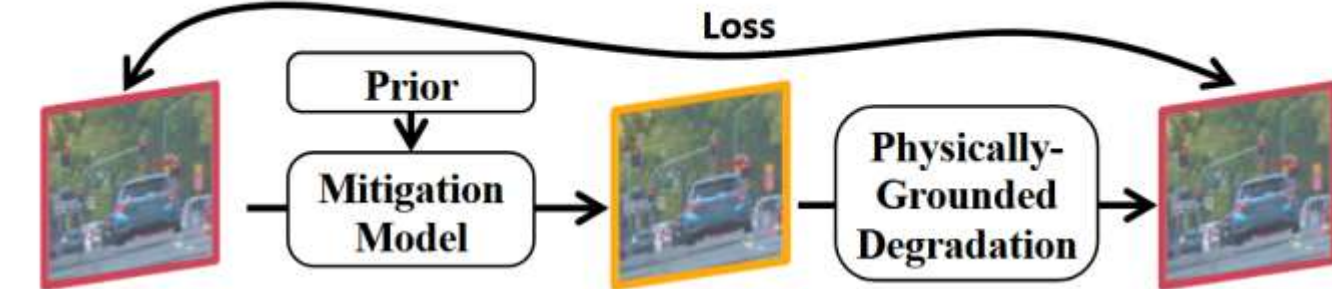
大气湍流由于温度梯度和其他能量源引起的空气混乱运动，导致视频中出现高频时间变化。这些变化表现为**随时间变化的倾斜和模糊**，偏离了真实情况，如湍流扭曲视频y-t切片中时间维度快速波动的图案。最先进的大气湍流缓解方法（DATUM）难以消除湍流带来的时间扭曲。虽然在消除湍流引起的空间扭曲（如模糊）方面相当有效，但不能有效消除**时间上的扭曲**。

现有解决方法

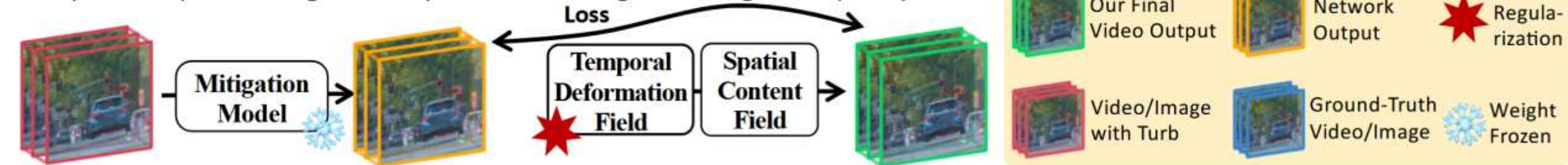
A. Supervised learning on paired video or image



B. self-supervised learning



C. Supervised pre-training + self-supervised learning on testing video (Ours)

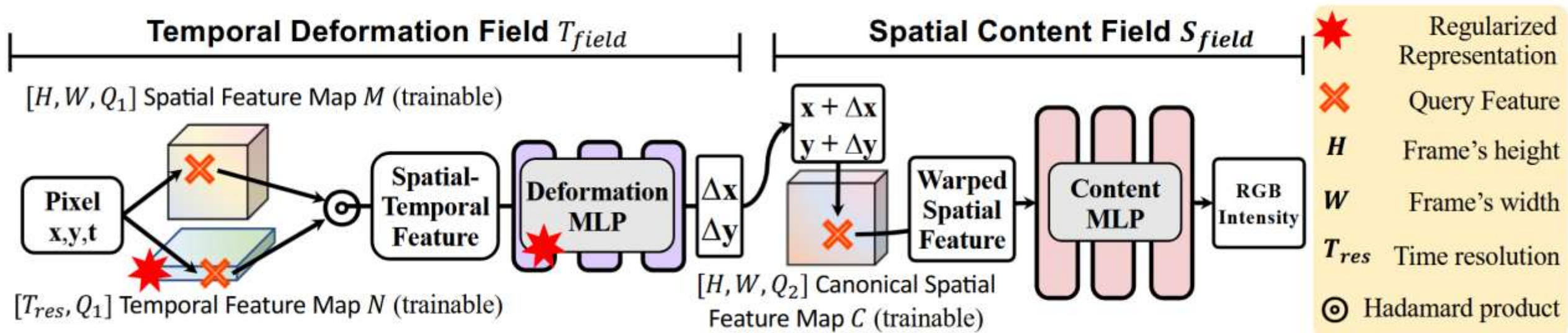


(a) 有监督的大气湍流缓解算法，难以实现到**真实世界的过渡**

(b) 自监督的大气湍流缓解算法，主要探索**静态场景**序列，从多个帧输出一个帧作为输入

(c) 本文提出方法先经过有监督的预训练，再自监督学习，专为**视频处理设计**

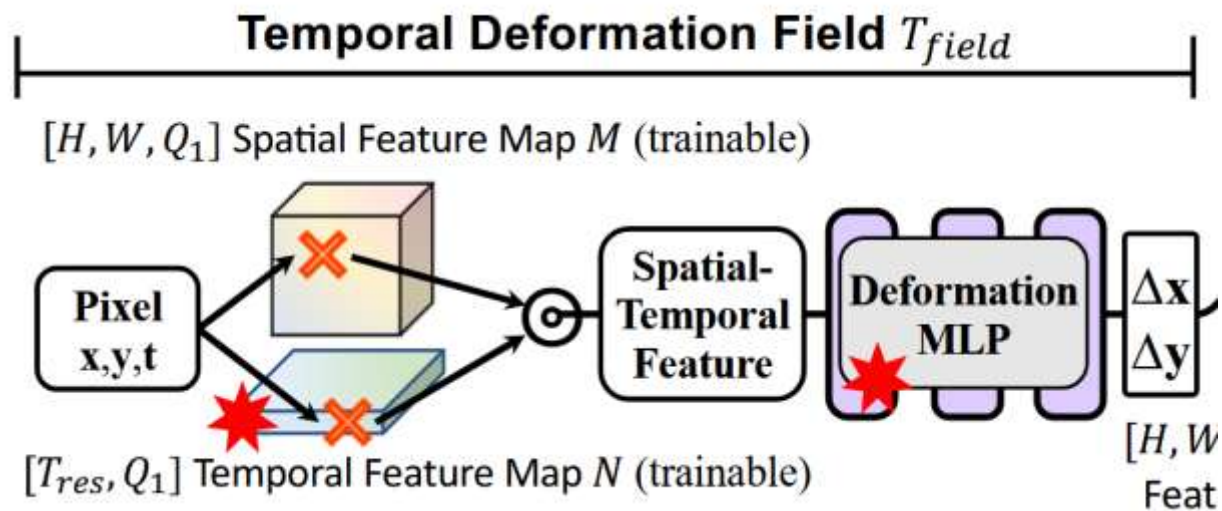
本文提出方法



本文提出ConVRT，将视频分解成两个隐式神经场：

- 时间变形场 **Temporal Deformation Field** (T_{field})
- 空间内容场 **Spatial Content Field** (S_{field})

• 时间变形场 **Temporal Deformation Field**

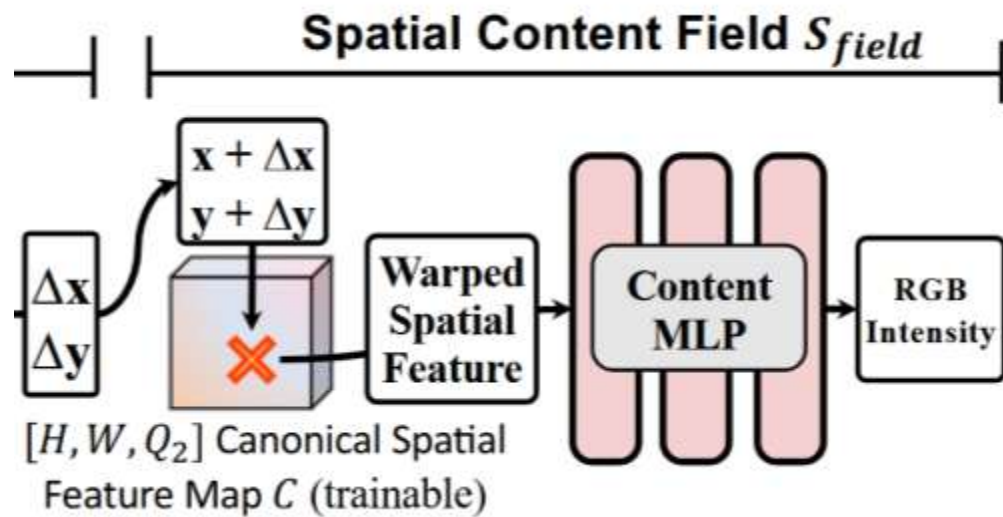


空间特征图 Spatial Feature Map M : $\mathbb{R}^{H \times W \times Q_1}$

时间特征图 Temporal Feature Map N : $\mathbb{R}^{T_{res} \times Q_1}$

时空特征 Spatial-Temporal Feature: $V_{x,y,t} = M_{x,y} \odot N_t$

- 空间内容场 **Spatial Content Field**



典范空间特征图 Canonical Spatial Feature Map $C: \mathbb{R}^{H \times W \times Q_2}$

空间矢量 spatial feature vector: $C_{x+\Delta x, y+\Delta y} \in \mathbb{R}^{Q_2}$

损失函数设置

ConVRT的损失分为两个部分，空间一致性损失 \mathcal{L}_{sim} 和时间一致性损失 \mathcal{L}_{temp}

$$\mathcal{L}_{total} = \mathcal{L}_{sim} + \lambda_{temp}\mathcal{L}_{temp}$$

空间一致性损失涉及到三种常用的图像质量损失：MSE损失，SSIM损失，LPIPS损失

$$\mathcal{L}_{sim} = \lambda_{mse}\mathcal{L}_{mse} + \lambda_{ssim}\mathcal{L}_{ssim} + \lambda_{lpips}\mathcal{L}_{lpips}$$

时间一致性损失涉及到一个惩罚不必要位移的 loss:

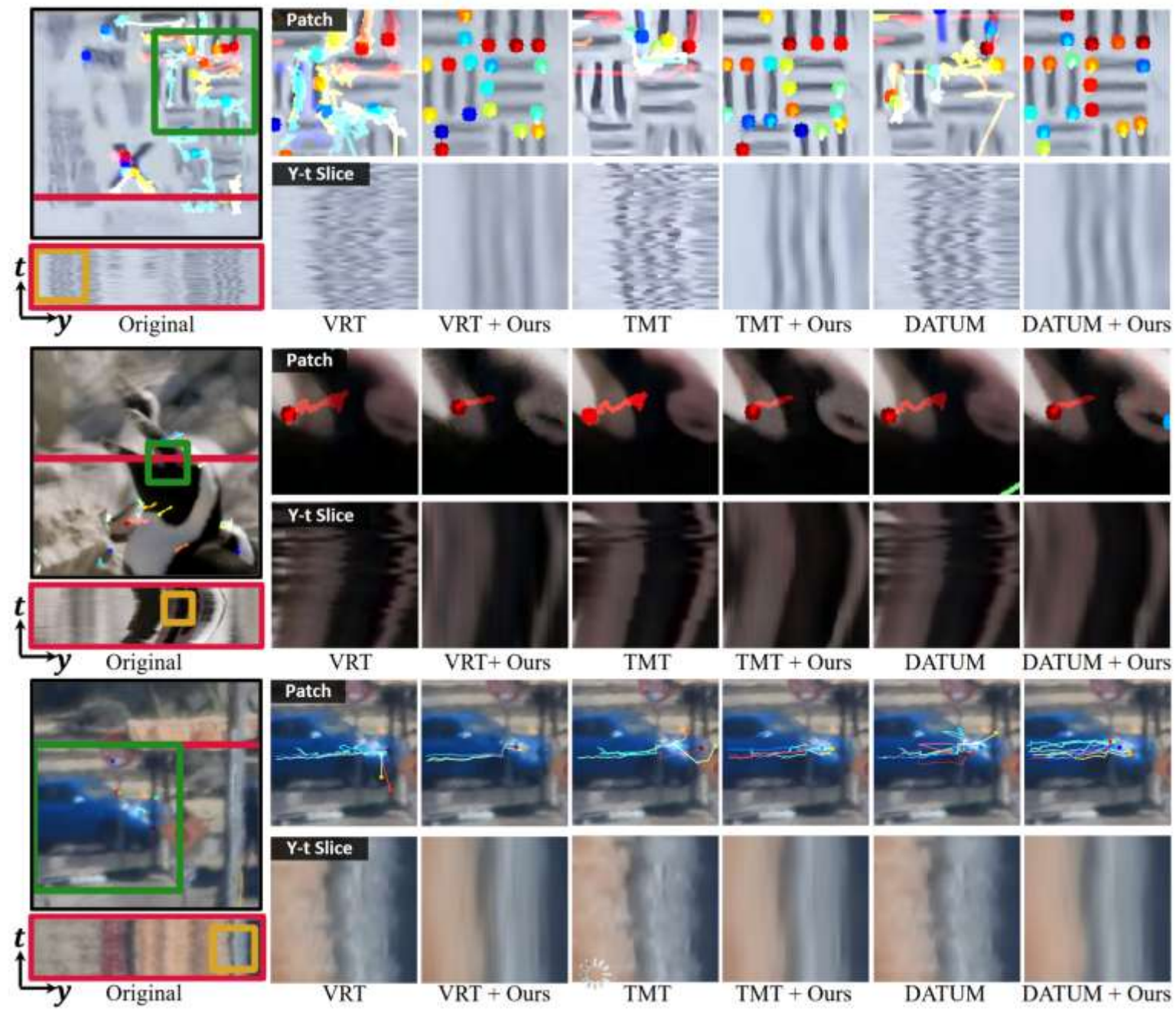
$$\mathcal{L}_{temp} = (1 - \text{Disparity}(I)) \cdot \|\text{Predicted Warp}\|_1$$

作者用 MiDaS^[1] 估计深度，得到 disparity，对于背景部分，权重更大，因为背景更加稳定，变化的一般是湍流，需要更大的惩罚，前景部分包含运动物体，设置更小的权重允许一定移动

Predicted Warp为 Δx , Δy 的变化量，希望使得总的变化量尽可能的少，说明湍流去除更加成功

[1] Ranftl R, Lasinger K, Hafner D, et al. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer[J]. IEEE transactions on pattern analysis and machine intelligence, 2020, 44(3): 1623-1637.

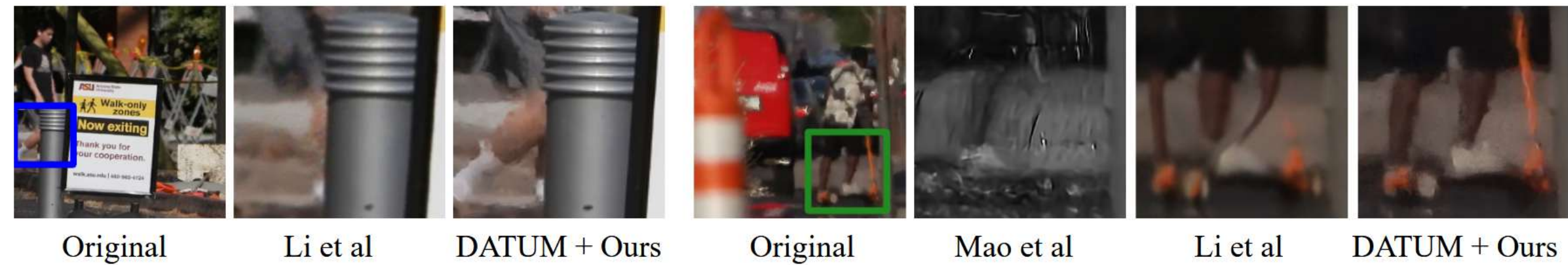
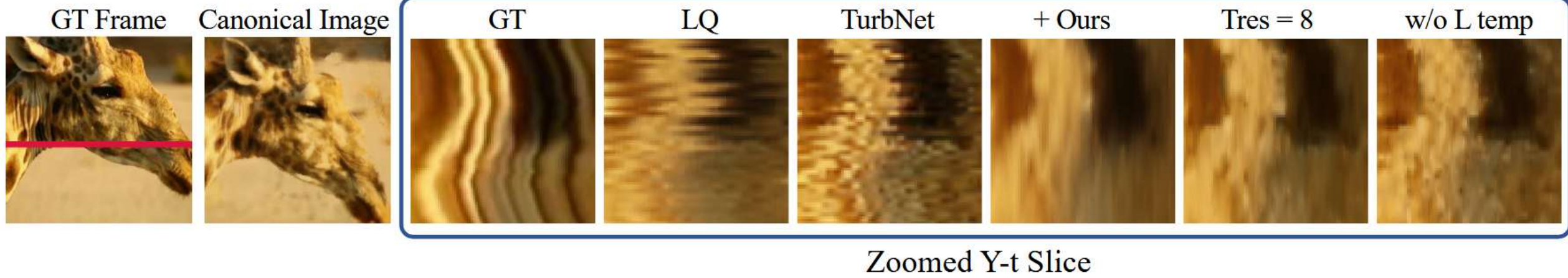
实验结果



实验结果

Turb Type	Dataset	Metrics	Video-based ATM						Image-based ATM			No Base Method			Other Restoration Method		
			TMT	+ ConVRT	Gain	DATUM	+ ConVRT	Gain	TurbNet	+ ConVRT	Gain	Ori	+ ConVRT	Gain	VRT	+ ConVRT	Gain
Real	HeatChamber(63)	$E_{warp} \downarrow$	24.21	19.78	-4.43	22.43	17.77	-4.66	41.48	17.36	-24.12	24.33	16.19	-8.14	25.77	16.19	-9.58
		$PSNR_{x-t} \uparrow$	18.45	18.60	+0.15	19.41	19.60	+0.19	18.40	19.33	+0.93	23.86	24.18	+0.32	23.84	24.18	+0.34
		$Flow_{tv} \downarrow$	5695.27	2786.99	-2908.28	5794.12	2509.21	-3284.91	17030.68	2383.29	-14647.39	9471.83	2154.55	-7317.28	9314.95	2154.55	-7160.40
		$PSNR \uparrow$	18.41	18.59	+0.18	19.25	19.46	+0.21	18.27	18.98	+0.71	19.79	19.96	+0.17	19.69	19.96	+0.27
		$SSIM \uparrow$	0.67	0.68	+0.01	0.69	0.70	+0.01	0.63	0.68	+0.05	0.67	0.68	+0.01	0.68	0.68	+0.01
	OTIS(63)	$Slice_{tv} \downarrow$	1365.77	387.40	-978.37	1237.91	365.09	-872.82	3124.37	638.03	-2486.34	1344.28	294.93	-1049.35	1579.10	313.47	-1265.63
		$Flow_{tv} \downarrow$	7334.87	963.53	-6371.34	6742.56	871.26	-5871.30	11454.92	811.78	-10643.15	7827.12	670.35	-7156.77	8985.64	662.76	-8322.87
	CLEAR (64)	$Slice_{tv} \downarrow$	115.34	109.71	-5.63	129.34	113.92	-15.42	377.62	210.78	-166.84	172.82	104.76	-68.06	186.76	105.31	-81.45
		$Flow_{tv} \downarrow$	3916.67	960.54	-2956.13	4023.44	933.97	-3089.47	11827.17	995.37	-10831.80	8333.30	845.42	-7487.88	9120.35	852.03	-8268.32
	TSRWGAN(4)	$Slice_{tv} \downarrow$	129.22	135.70	+6.48	123.32	124.93	+1.61	523.47	311.07	-212.40	151.65	115.07	-36.58	168.90	118.22	-50.68
		$Flow_{tv} \downarrow$	2176.51	419.36	-1757.15	2279.42	411.80	-1867.61	6038.92	474.29	-5564.63	3460.89	394.22	-3066.67	3700.43	393.54	-3306.89

实验结果



实验结果

