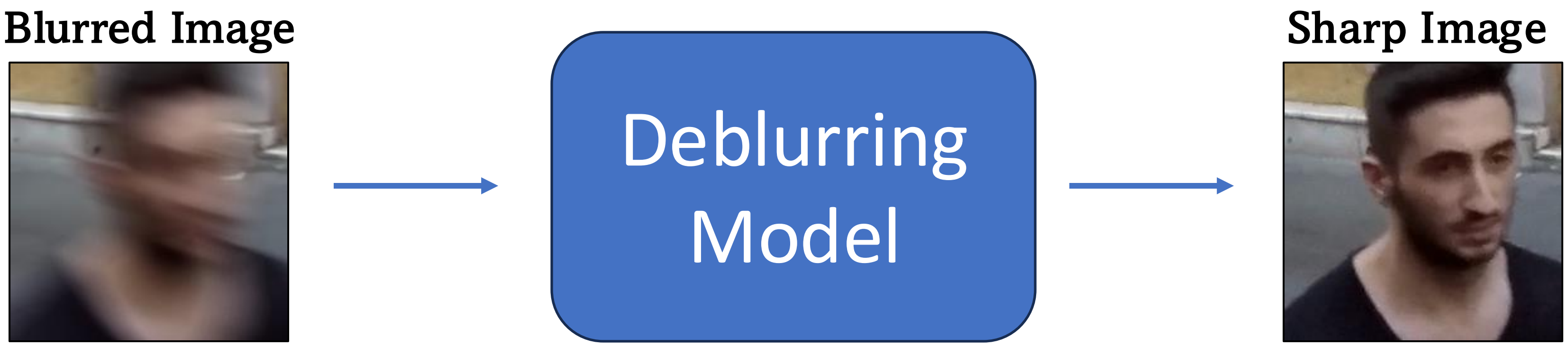
BlurDM: A Blur Diffusion Model for Image Deblurring

Jin-Ting He, Fu-Jen Tsai, Yan-Tsung Peng, Min-Hung Chen, Chia-Wen Lin, and Yen-Yu Lin
National Yang Ming Chiao Tung University, National Tsing Hua University,
National Chengchi University, NVIDIA

Introduction - Task

Dynamic Scene Image Deblurring



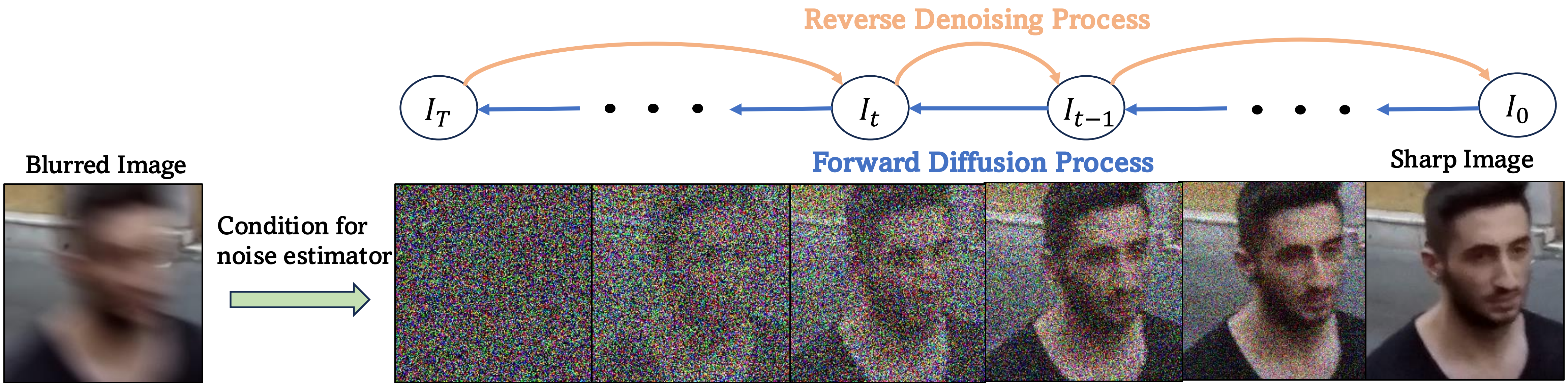
Introduction - Previous works

Existing regression-based image deblurring

- Building on CNNs or Transformers

Existing diffusion-based image deblurring

- Directly adopting the diffusion framework to generate sharp image



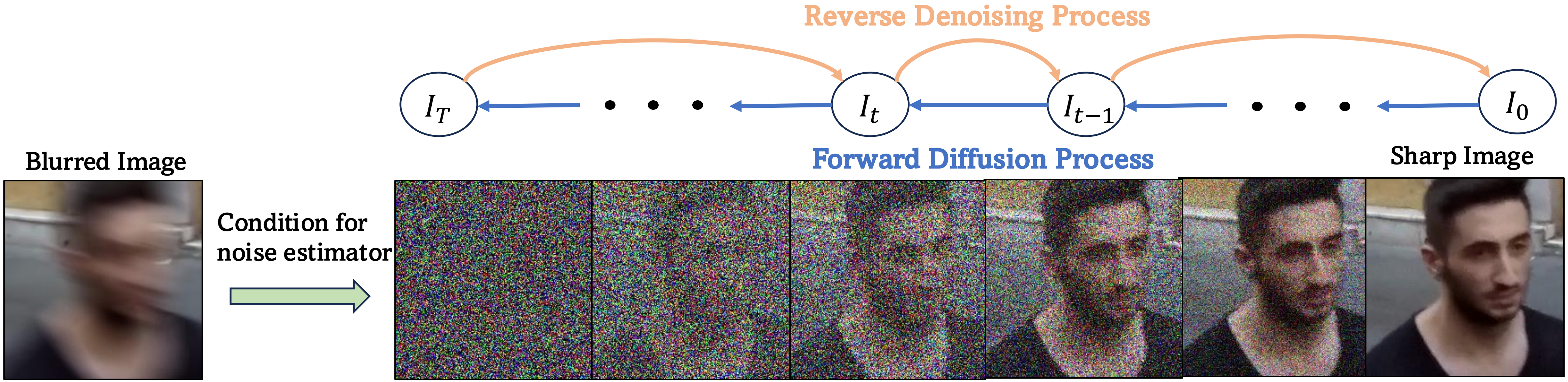
Introduction - Previous works

Existing regression-based image deblurring

- Building on CNNs or Transformers
- Producing over-smoothed results due to their reliance on regression losses

Existing diffusion-based image deblurring

- Directly adopting the diffusion framework to generate sharp image



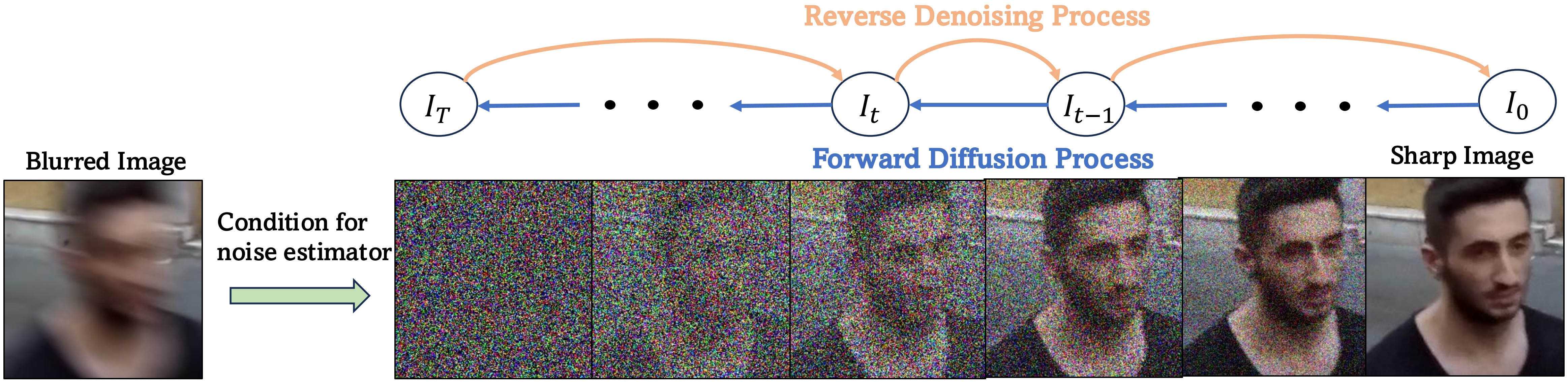
Introduction - Previous works

Existing regression-based image deblurring

- Building on CNNs or Transformers
- Producing over-smoothed results due to their reliance on regression losses

Existing diffusion-based image deblurring

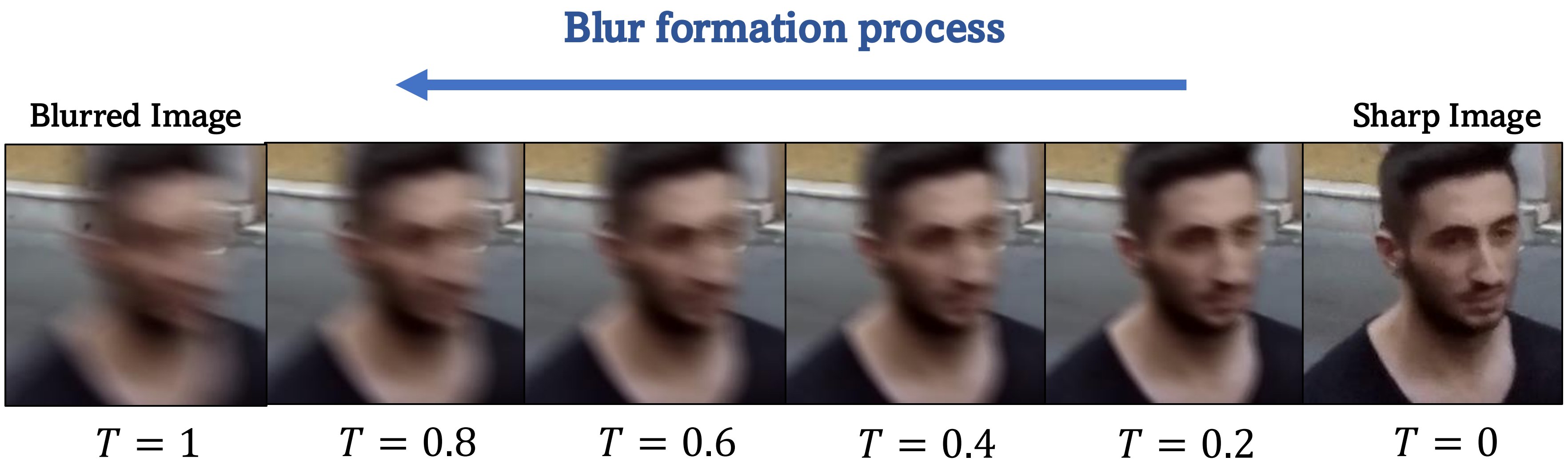
- Directly adopting the diffusion framework to generate sharp image
- Without considering the blur characteristics, leading to limited deblurring results



Introduction - Observations

Blur formation process

- Blur intensity progressively accumulates along the motion trajectory during the exposure process

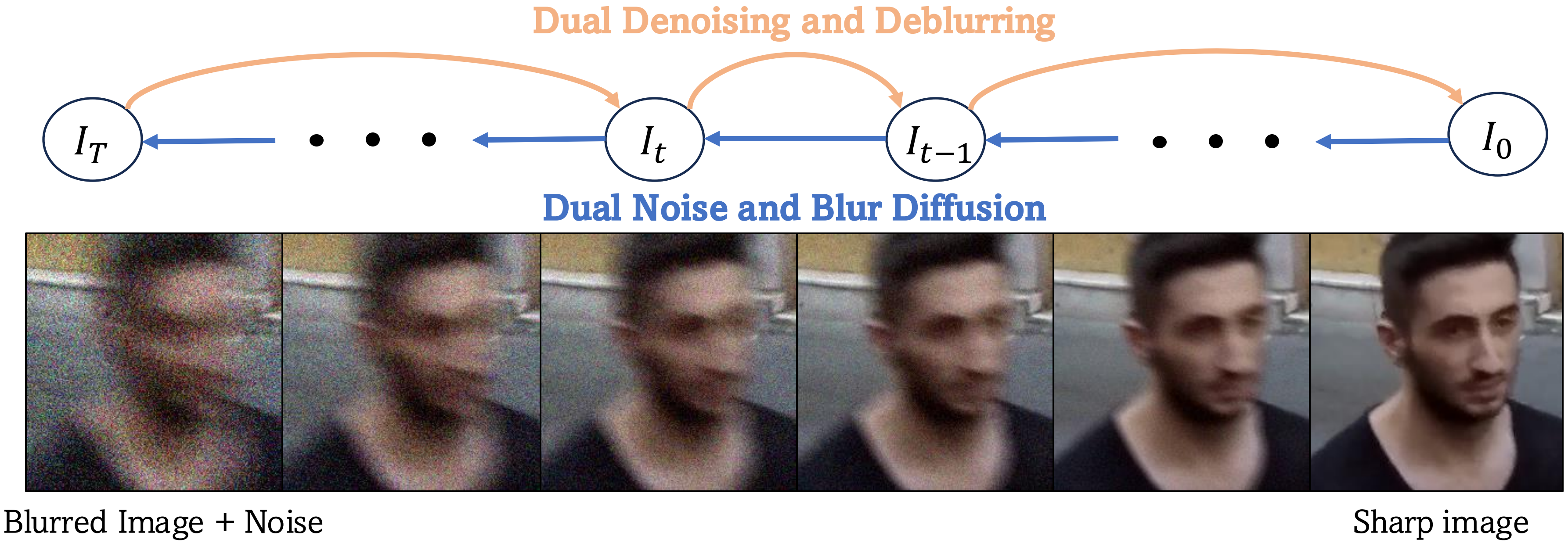


T is the additional exposure time

Introduction - This paper

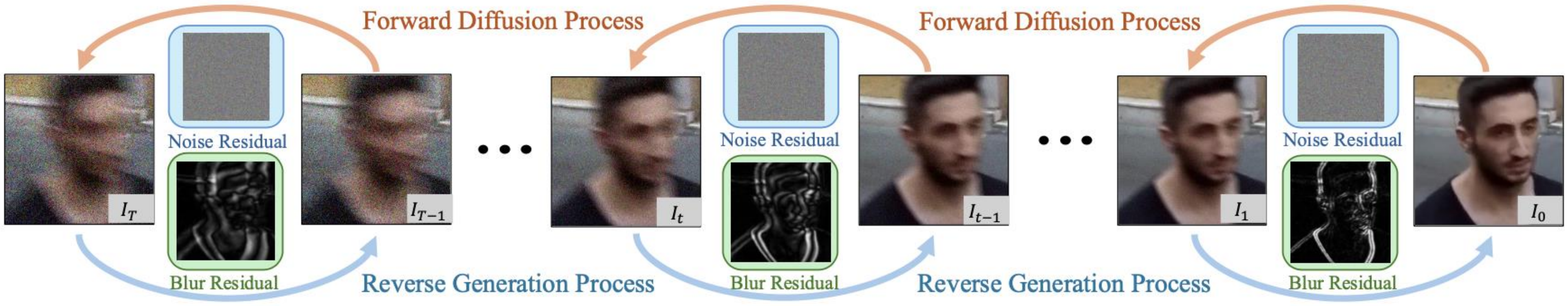
Blur Diffusion Model (BlurDM)

- Aligning the diffusion process with the physical principles of blur formation process
- **Dual Noise and Blur Diffusion (forward):** Adding both noise and blur to sharp images to generate blurred and noisy images
- **Dual Denoising and Deblurring (reverse):** Removing both noise and blur to retore sharp images



Method – Blur Diffusion Model

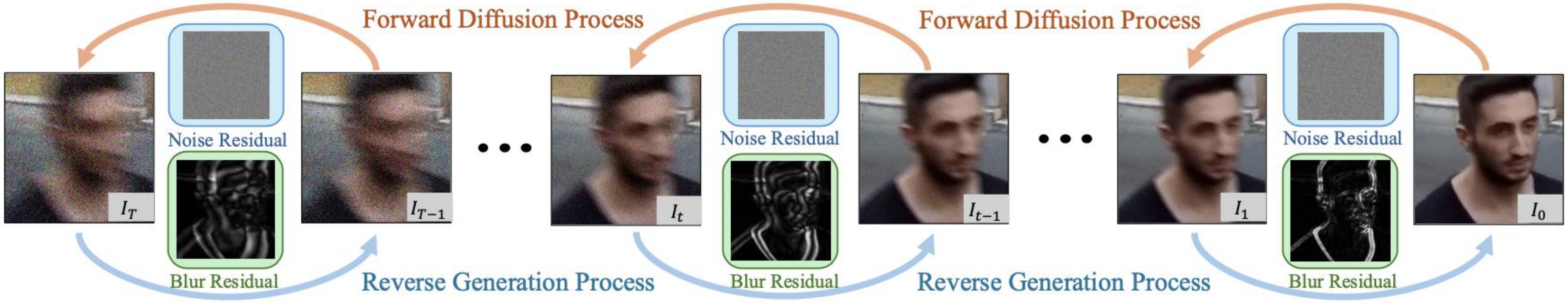
How we progressively add/remove the "blur" in each time step?



Method – Blur Diffusion Model

How we progressively add/remove the "blur" in each time step?

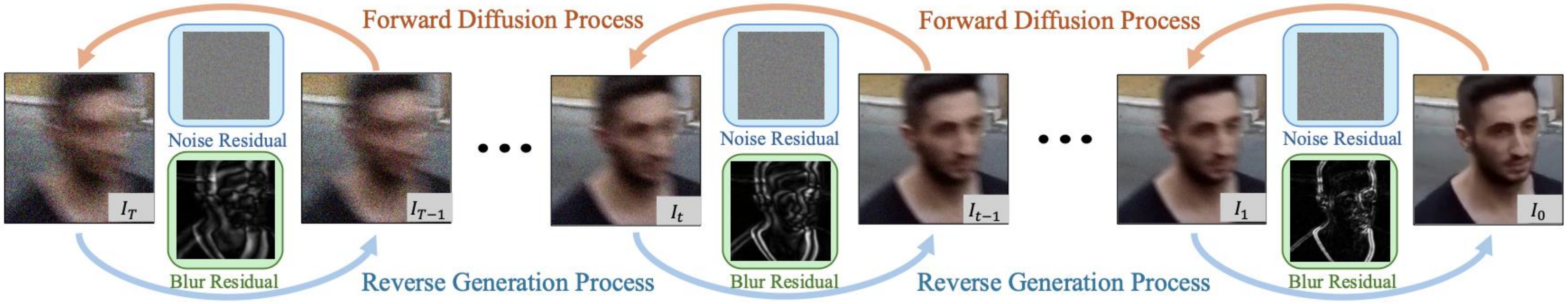
- **Blur Residual:** the variation in blur across different exposure durations



Method – Blur Diffusion Model

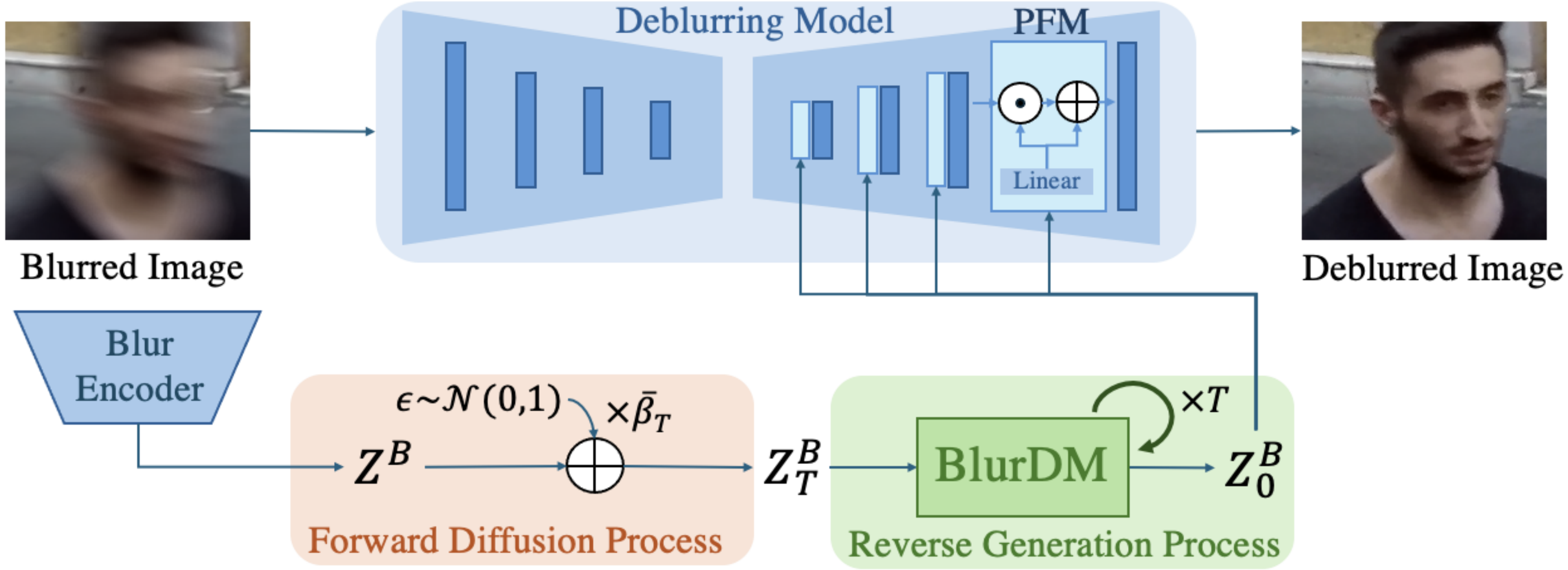
Dual Noise and Blur Diffusion (forward) : Adding both noise residual and blur residual to sharp images to generate blurred and noisy images

Dual Denoising and Deblurring (reverse) : Removing both noise residual and blur residual to restore sharp images through **Noise Residual Estimator** and **Blur Residual Estimator**



Method – Latent Blur Diffusion Model

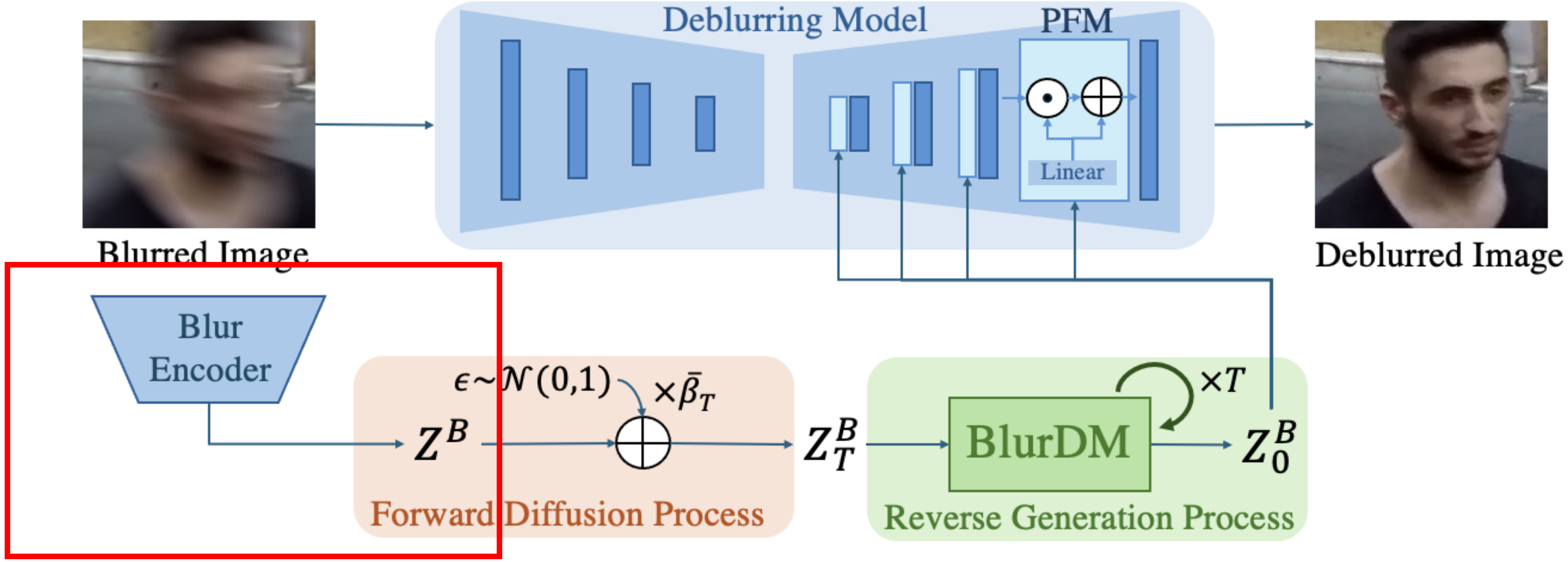
We adopt our BlurDM into latent space to generate the sharp prior for improving the existing deblurring model



Method – Latent Blur Diffusion Model

We adopt our BlurDM into latent space to generate the sharp prior for improving the existing deblurring model

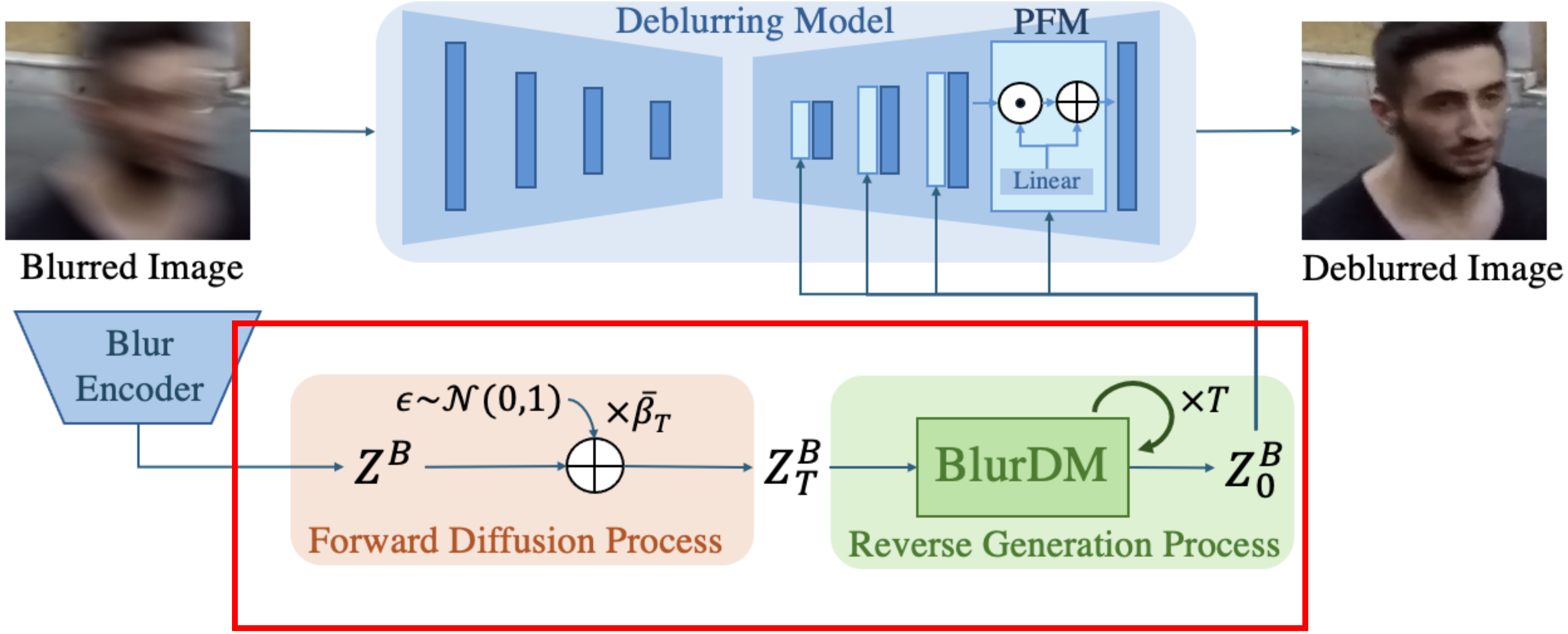
1. Encoding the blurred image into a latent feature vector using a Blur Encoder



Method – Latent Blur Diffusion Model

We adopt our BlurDM into latent space to generate the sharp prior for improving the existing deblurring model

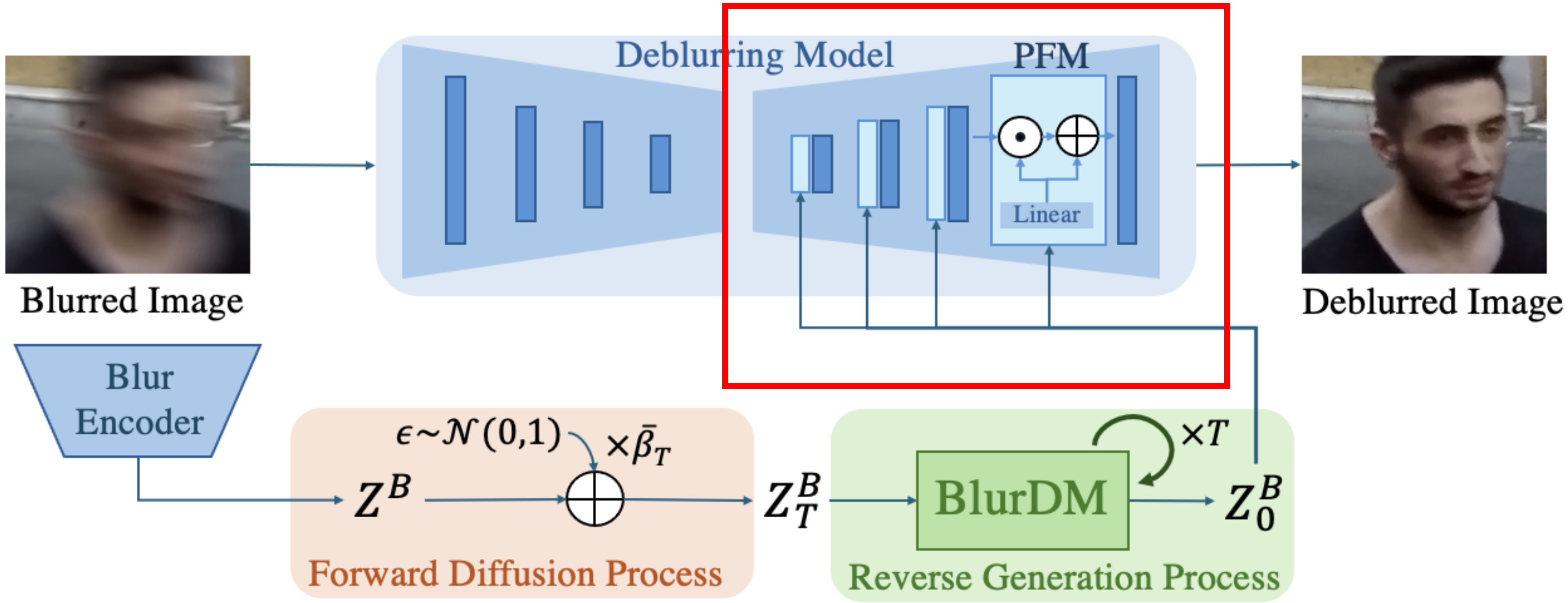
1. Encoding the blurred image into a latent feature vector using a Blur Encoder
2. Applying BlurDM to generate the sharp prior from this feature vector



Method – Latent Blur Diffusion Model

We adopt our BlurDM into latent space to generate the sharp prior for improving the existing deblurring model

1. Encoding the blurred image into a latent feature vector using a Blur Encoder
2. Applying BlurDM to generate the sharp prior from this feature vector
3. Fusing the sharp prior into each decoder of the deblurring network



Experiments

Baseline: Deblurring performance without BlurDM

BlurDM: Deblurring performance with BlurDM

Method		GoPro			HIDE			RealBlur-J			RealBlur-R		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
MIMO-UNet	Baseline	32.44	0.957	0.0115	30.00	0.930	0.0217	31.59	0.918	0.0345	39.03	0.968	0.0215
	BlurDM	**32.93**	**0.961**	**0.0091**	**30.73**	**0.939**	**0.0168**	**32.13**	**0.926**	**0.0264**	**39.63**	**0.972**	**0.0172**
Stripformer	Baseline	33.09	0.962	0.0085	31.03	0.940	0.0147	32.48	0.929	0.0222	39.84	0.974	0.0138
	BlurDM	**33.53**	**0.966**	**0.0074**	**31.36**	**0.944**	**0.0122**	**33.53**	**0.938**	**0.0175**	**41.00**	**0.977**	**0.0115**
FFTformer	Baseline	34.21	0.969	0.0067	31.62	0.946	0.0153	32.62	0.933	0.0220	40.11	0.973	0.0149
	BlurDM	**34.34**	**0.970**	**0.0060**	**31.76**	**0.947**	**0.0145**	**32.92**	**0.939**	**0.0195**	**40.55**	**0.975**	**0.0136**
LoFormer	Baseline	33.54	0.966	0.0084	31.18	0.943	0.0176	32.23	0.932	0.0223	40.36	0.974	0.0148
	BlurDM	**33.70**	**0.967**	**0.0073**	**31.27**	**0.944**	**0.0158**	**33.47**	**0.941**	**0.0189**	**40.92**	**0.976**	**0.0127**
Average Gain		**+0.31**	**+0.003**	**-0.0013**	**+0.32**	**+0.004**	**-0.0025**	**+0.78**	**+0.008**	**-0.0047**	**+0.69**	**+0.003**	**-0.0025**

Experiments

Baseline: Deblurring results without BlurDM

BlurDM: Deblurring results with BlurDM

Blurred Image | Blurred Patch | Baseline | BlurDM | Sharp Patch

Conclusion

- We present BlurDM, a novel diffusion-based network that incorporates the blur formation to enhance dynamic scene image deblurring
- We propose a dual noise and blur diffusion process and derive a dual denoising and deblurring formulation, which allows BlurDM to implicitly estimate blur residuals instead of relying on the ground truth
- Extensive experiments demonstrate that BlurDM significantly and consistently improves four deblurring models on four benchmark datasets

Source code

2025

BlurDM

Thank You For Listening