Supplementary Materials for the Paper VQCNIR: Clearer Night Image Restoration with Vector-Quantized Codebook

Wenbin Zou,[1] Hongxia Gao†,[1,2*] Tian Ye,[3] Liang Chen,[4]
Weipeng Yang,[1] Shasha Huang,[1] Hongshen Chen,[1] Sixiang Chen[3]
[1]The School of Automation Science and Engineering, South China University of Technology,
[2]Research Center for Brain-Computer Interface, Pazhou Laboratory, Guangzhou,
[3]The Hong Kong University of Science and Technology, Guangzhou,
[4] College of Photonic and Electronic Engineering, Fujian Normal University
†Email: hxgao@scut.edu.cn

In this supplementary material, we first introduced the architectural details of our proposed Clearer Night Image Restoration with Vector-Quantized Codebook (VQCNIR). Then, we further provided the objective function for training the high-quality Vector Quantized (VQ) codebook, as well as the training dataset and details. Next, we further analyzed and discussed each module of our proposed VQCNIR network. Finally, we provided more visual comparisons on LOL-Blur and real-world images.

Network Architectures of VQCNIR

In complementary to the simple network architecture in the paper, we provide details of hyperparameters for the VQCNIR encoder and parallel decoder in Table 1. In our experiment, we use the codebook $\mathcal{Z}$ with size 1024×512. The input image is of size 256×256 and downsampled into 32×32 feature maps. Each level of the encoder comprises two residual blocks, with Adaptive Illumination Enhancement Module (AIEM) applied at the encoder output to enhance feature illumination. For decoder G, each level contains two residual blocks. In decoder D, each level consists of one residual block followed by our proposed Deformable Bi-Directional Cross-Attention (DBCA) module. We use a 3×3 convolution to downsample the image and bilinear upsampling to recover the image resolution. VQCNIR has 45.9M params and 650.55 GFlops on 512^2 image.

Training Details of VQ Codebook

Training Objectives of VQ codebook

Since the Vector Quantization operation is non-differentiable, we follow VQGAN and use the straight-through gradient estimator (Esser, Rombach, and Ommer 2021), which simply copies the gradients from the decoder G into the encoder E. This allows the model and codebook to be trained end-to-end via the loss function $\mathcal{L}_{code}$. We adopt the pixel-wise construction loss $\mathcal{L}_{rec}$ for basic pixel reconstruction. Moreover, we used perceptual loss $\mathcal{L}_{pre}$, adversarial loss $\mathcal{L}_{adv}$, and semantic guided loss $\mathcal{L}_{SG}$ to improve the quality of the codebooks.

Pixel Reconstruction Loss: We utilize the L_1 loss in the sRGB space as the pixel-wise construction loss, which can be denoted as:

$$\mathcal{L}_{rec} = ||\hat{x}_h - x_h||_1, \tag{1}$$

where $\hat{x}_h$ and x_h represent the reconstruction results and the input high-quality images, respectively. $|| \cdot ||_1$ denotes the L_1-norm.

Codebook Loss: To optimize the codebook, we use codebook loss to minimize the distance between the codebook and input feature embeddings and then update the codebook:

$$\mathcal{L}_{code} = ||\text{sg}[E(x_h)] - z_{\mathbf{q}}||_2^2 + \beta||\text{sg}[z_{\mathbf{q}}] - E(x_h)||_2^2, \tag{2}$$

where $\text{sg}[\cdot]$ and $E(\cdot)$ represent stop gradient operation and encoder E. $z_{\mathbf{q}}$ denotes the quantized features, and $\beta = 0.25$ according to (Esser, Rombach, and Ommer 2021).

Adversarial Loss: Similar to VQGAN, we incorporate an adversarial loss term to enhance the texture quality of the reconstructed output $\hat{x}_h$:

$$\mathcal{L}_{adv} = \log D(x_h) + \log(1 - D(\hat{x}_h)), \tag{3}$$

where $D(\cdot)$ indicates discriminator.

Semantic Guided Loss: The codebook is obtained via gradient descent, where patterns with similar visual appearances are grouped together irrespective of their semantics. To ensure that the codebook embedding maintains consistency between semantic information and textures, we introduce a regularization term that leverages perceptual features containing rich semantic information. This is accomplished by incorporating a semantic guidance loss term, following the approach of FeMaSR (Chen et al. 2022a):

$$\mathcal{L}_{SG} = ||Conv(z_{\text{q}}) - \phi(x_h)||_2^2, \tag{4}$$

where $Conv(\cdot)$ and $\phi(\cdot)$ denote the point-wise convolution and the pre-trained VGG19 network.

Perceptual Loss: We use the widely used perceptual loss (Johnson, Alahi, and Fei-Fei 2016) for the decoder output $\hat{x}_h$:

$$\mathcal{L}_{pre} = ||\phi(\hat{x}_h) + \phi(x_h)||_1, \tag{5}$$

where $\phi(\cdot)$ denotes the pre-trained VGG19 (Simonyan and Zisserman 2014) network. Finally, the train objectives of VQGAN are as follows:

$$\mathcal{L}_{vq} = \lambda_r\mathcal{L}_{rec} + \lambda_c\mathcal{L}_{code} + \lambda_a\mathcal{L}_{adv} + \lambda_s\mathcal{L}_{SG} + \lambda_p\mathcal{L}_{per}, \tag{6}$$

where λ_r, λ_c, λ_a, λ_s, and λ_p denote the hyperparameters of each loss function, respectively.

*Corresponding author.

Table 1: The detailed architecture of VQCNIR. The residual block consists of GN-3×3 Conv-Act-GN-Act-3×3 Conv. s: stride in convolution; g: groups in GroupNorm (GN); c: channels; f: compression patch size; f*: features after illumination enhancement.

Input Size	Encoder	Decoder D	Decoder G
f1:256×256	Conv3×3, 2-s, 64-c→128-c	Conv3×3, 64-c→3-c	Conv3×3, 64-c→3-c
	{Residual block, 128-c, 32-g}×2		
f2:128×128	Conv3×3, 2-s, 128-c→256-c	Bilinear upsampling 2×	Bilinear upsampling 2×
	{Residual block, 256-c, 32-g}×2	DBCA, 64-c	Conv3×3, 128-c
		{Residual block, 128-c→64-c, 32-g} ×1	{Residual block, 64-c, 32-g} × 2
f4:64×64	Conv3×3, 2-s, 256-c	Bilinear upsampling 2×	Bilinear upsampling 2×
	{Residual block, 256-c, 32-g}×2	DBCA, 128-c	Conv3×3, 256-c
		{Residual block, 256-c→128-c, 32-g} ×1	{Residual block, 128-c, 32-g} × 2
f8:32×32	Conv3×3, 2-s, 256-c	Bilinear upsampling 2×	Bilinear upsampling 2×
	{Residual block, 256-c, 32-g}×2	DBCA, 256-c	Conv3×3, 256-c
		{Residual block, 512-c→256-c, 32-g} ×1	{Residual block, 256-c, 32-g} × 2
f*8:32×32	{AIEM, 256-c)×12	DBCA, 256-c	-

Dataset and Training Details

To obtain high-quality codebooks, we follow the same setup as in the previous FeMaSR (Chen et al. 2022a), using DIV2K (Timofte et al. 2017) and Flickr2K (Timofte et al. 2017) for training VQ Codebook. Moreover, we use the Adam (Kingma and Ba 2014) optimizer with a learning rate of 1×10^{-4} to train the optimized VQGAN to obtain the trained VQ codebook. For data enhancement, we cropped the input image to a random size of 256×256. where λ_r, λ_c, λ_a, λ_s, and λ_p hyperparameters in the loss function is set to $\{1, 1, 1, 0.25\}$ respectively.

More Discussions on VQCNIR

The Effectiveness of AIEM.

To further validate the efficacy of our proposed AIEM, we conducted ablation experiments using ResBlock, NAFBlock, and SwinTransformer respectively. As Table 2 shows, ResBlock and NAFBlock exhibit poor performance owing to limited receptive fields. Although SwinTransformer can leverage strong non-local correlations for better results, it is computationally intensive. In light of these factors, our AIEM design utilizing channel correlations to estimate illumination curves avoids computational overload and maintains interpretability, achieving optimal performance.

Visual comparisons between different modules are provided in Figure 1, including ResBlock, NAFBlock, and SwinTransformer. As depicted, ResBlock and NAFBlock still suffer from blurring and insufficient illumination due to restricted receptive fields. SwinTransformer can effectively reduce blurring through powerful long-range modeling, but lacks explicit illumination enhancement, resulting in artifacts and inadequate lighting improvement. Our method estimates illuminance curve parameters via feature channel dependencies, thereby enhancing illumination. Consequently, our approach recovers textures and details closer to the ground truth image, as evident in the figure.

Table 2: Ablation studies on Adaptive Illumination Enhancement Module.

method	PSNR	SSIM	LPIPS
ResBlock (2017)	26.35	0.8483	0.1089
NAFBlock (2022b)	26.71	0.8593	0.1032
SwinTransformer (2021)	27.34	0.8668	0.1013
Ours	27.79	0.8750	0.0960

The Effectiveness of DBCA.

We also implemented a series of ablation experiments to verify the effectiveness of DBCA and the results are shown in Table 3. We employ 4 different feature fusion methods separately to verify the effectiveness of our proposed method. As can be seen from the table, simply fusing the features in the decoder does not result in a performance gain. Therefore, the balance between degraded features and high-quality prior features is important. Adaptive fusion mechanisms such as SKFF and cross-attention assign weights directly without considering the differences between degenerate features and high-quality priors, making it impossible to obtain the best performance. Our proposed DBCA utilizes bi-directional cross-attention and deformable convolution to effectively fuse these two features and therefore obtain the best performance.

To demonstrate the effectiveness of our proposed DBCA, we provide visuals of the model with different feature fusion methods. As can be seen in Figure 2, simple fusion methods such as add and concat are not effective in reducing the blurring effect in the image. SKFF and Cross-Attention can achieve relatively good deblurring performance. However, these methods still have blurring and artifacts on fine textures. As can be seen from the figure, our proposed DBCA method results in a sharper texture.

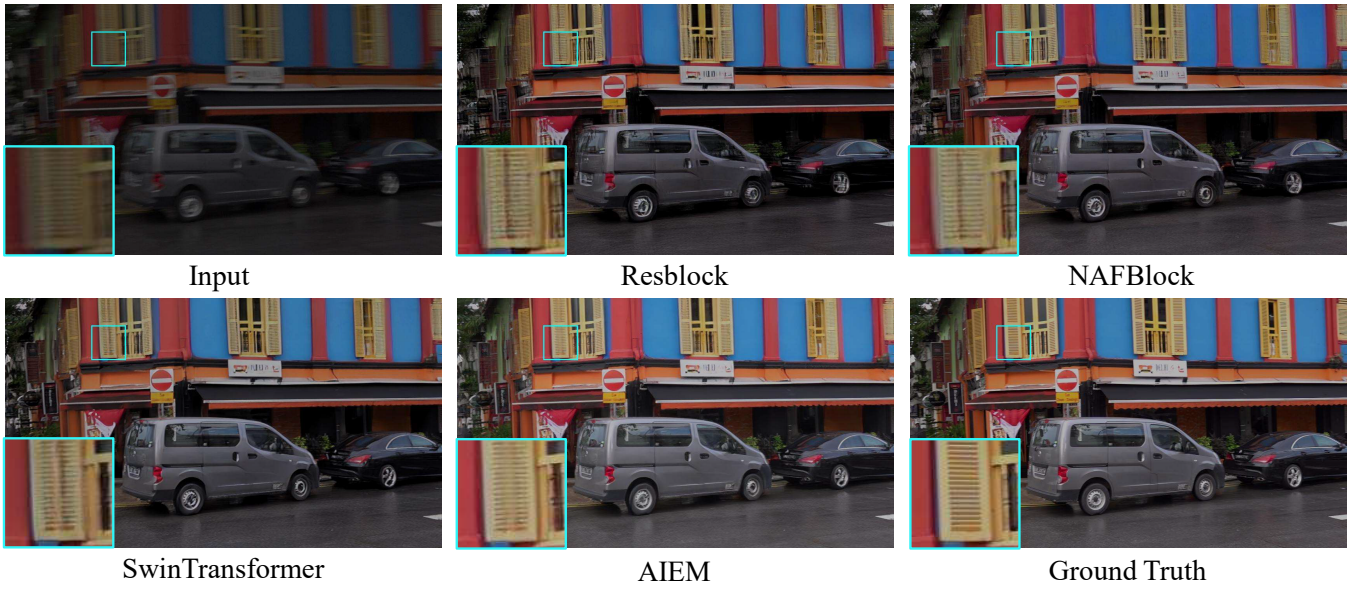

Figure 1: Visual result comparison of different illumination enhancement models: Resblock, NAFBlock, SwinTransformer, and AIEM. (Zoom in for the best view)

Table 3: Ablation studies on parallel decoder with different feature fusion.

method	PSNR	SSIM	LPIPS
Add	27.15	0.8670	0.1007
Concat	27.26	0.8653	0.1008
SKFF (2020)	27.48	0.8692	0.1001
Cross-attention (2021)	27.61	0.8748	0.0992
Ours	27.79	0.8750	0.0960

More Results

In this section, we present more visual comparisons with the baselines studied in the main manuscript: Restormer (Zamir et al. 2022) $\rightarrow$ LLFlow (Wang et al. 2022a), Uformer (Wang et al. 2022b) $\rightarrow$ LLFlow (Wang et al. 2022a), LLFlow (Wang et al. 2022a) $\rightarrow$ Restormer (Zamir et al. 2022), DMPHN* (Zhang et al. 2019), MIMO* (Cho et al. 2021), Restormer* (Zamir et al. 2022), LLFlow* (Wang et al. 2022a), and LED-Net (Zhou, Li, and Change Loy 2022). Figure 3 provides more visual comparisons on our LOL-Blur Dataset. In addition, Figure 4 provides more visual comparisons of real-world night blurry images. To demonstrate the generalizability in the wild of our dataset and network, we also test on more real-world night blurry images in the RealBlur dataset (Rim et al. 2020). Figure 5 and 6 provide more results in different scenarios on the RealBlur dataset.

References

Chen, C.; Shi, X.; Qin, Y.; Li, X.; Han, X.; Yang, T.; and Guo, S. 2022a. Real-world blind super-resolution via feature matching with implicit high-resolution priors. In *Proceedings of the 30th ACM International Conference on Multimedia*, 1329–1338.

Chen, C.-F. R.; Fan, Q.; and Panda, R. 2021. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, 357–366.

Chen, L.; Chu, X.; Zhang, X.; and Sun, J. 2022b. Simple baselines for image restoration. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, 17–33. Springer.

Cho, S.-J.; Ji, S.-W.; Hong, J.-P.; Jung, S.-W.; and Ko, S.-J. 2021. Rethinking coarse-to-fine approach in single image deblurring. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4641–4650.

Esser, P.; Rombach, R.; and Ommer, B. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12873–12883.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, 694–711. Springer.

Kingma, D.; and Ba, J. 2014. Adam: A Method for Stochastic Optimization. *Computer ence*.

Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Van Gool, L.; and Timofte, R. 2021. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1833–1844.

Lim, B.; Son, S.; Kim, H.; Nah, S.; and Mu Lee, K. 2017. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 136–144.

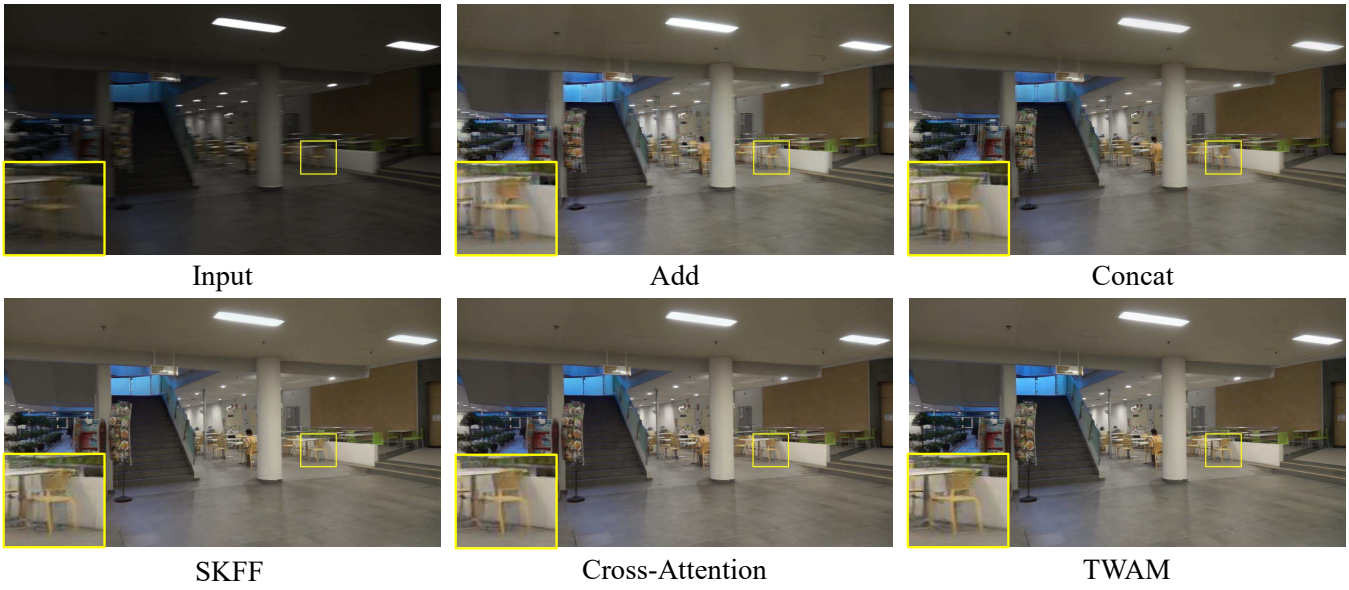

Figure 2: Visual result comparison of different feature fusion methods: Add, Concat, SKFF, Cross-Attention, and DBCA. (Zoom in for the best view)

Rim, J.; Lee, H.; Won, J.; and Cho, S. 2020. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, 184–201. Springer.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Timofte, R.; Agustsson, E.; Gool, L. V.; Yang, M.; Zhang, L.; Lim, B.; Son, S.; Kim, H.; Nah, S.; and et al., K. M. L. 2017. NTIRE 2017 Challenge on Single Image Super-Resolution: Methods and Results. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1110–1121.

Timofte, R.; Agustsson, E.; Van Gool, L.; Yang, M.-H.; Zhang, L.; Lim, B.; et al. 2017. NTIRE 2017 Challenge on Single Image Super-Resolution: Methods and Results. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Wang, Y.; Wan, R.; Yang, W.; Li, H.; Chau, L.-P.; and Kot, A. 2022a. Low-light image enhancement with normalizing flow. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 2604–2612.

Wang, Z.; Cun, X.; Bao, J.; Zhou, W.; Liu, J.; and Li, H. 2022b. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 17683–17693.

Zamir, S. W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F. S.; and Yang, M.-H. 2022. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5728–5739.

Zamir, S. W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F. S.; Yang, M.-H.; and Shao, L. 2020. Learning enriched features for real image restoration and enhancement. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, 492–511. Springer.

Zhang, H.; Dai, Y.; Li, H.; and Koniusz, P. 2019. Deep stacked hierarchical multi-patch network for image deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5978–5986.

Zhou, S.; Li, C.; and Change Loy, C. 2022. Lednet: Joint low-light enhancement and deblurring in the dark. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VI*, 573–589. Springer.

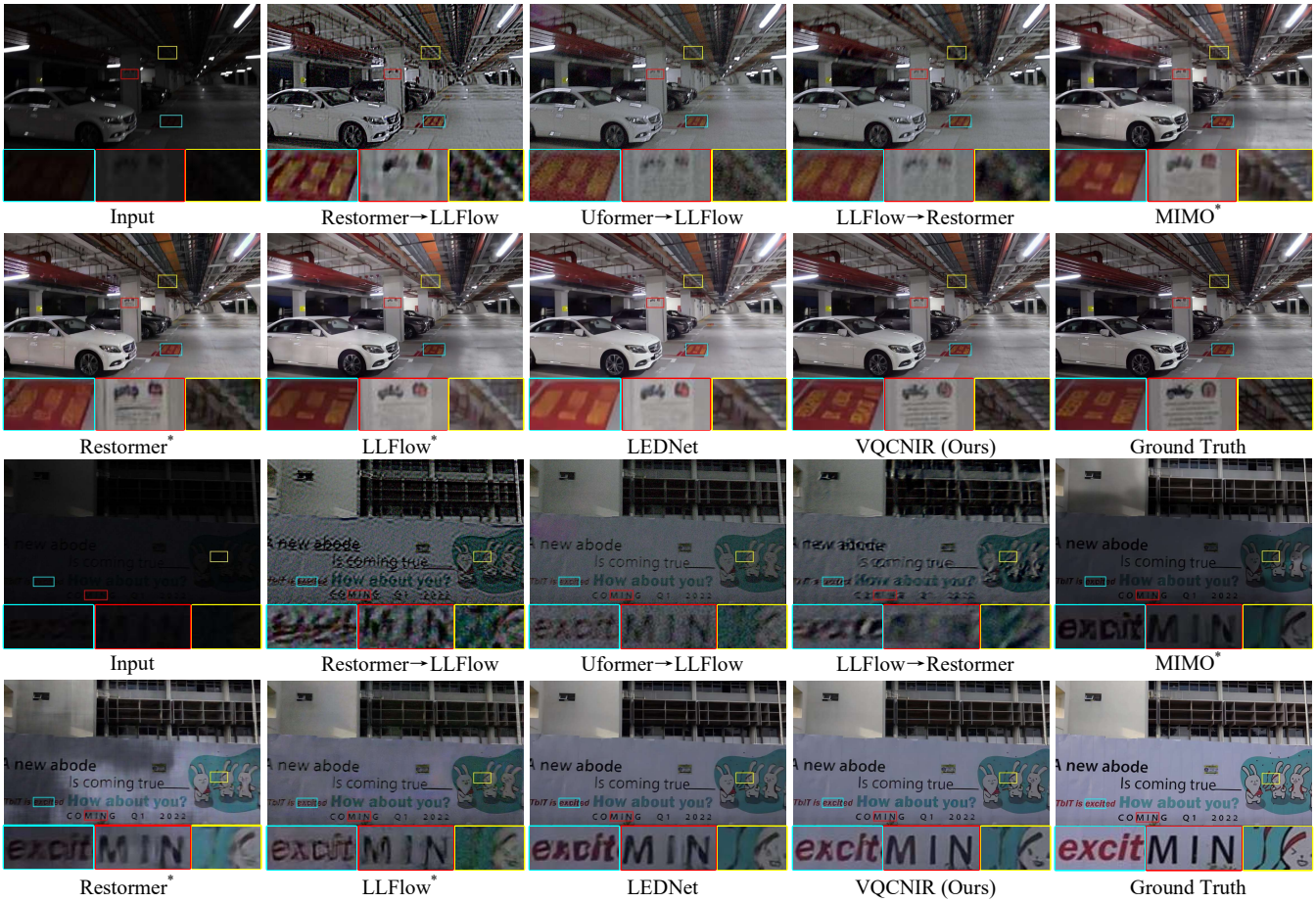

Figure 3: Visual comparison on our LOL-Blur dataset (Zhou, Li, and Change Loy 2022). The proposed VQCNIR generates much sharper images with visually pleasing results. The '*' indicates the network is trained with our LOL-Blur dataset (Zhou, Li, and Change Loy 2022). (Zoom in for the best view)

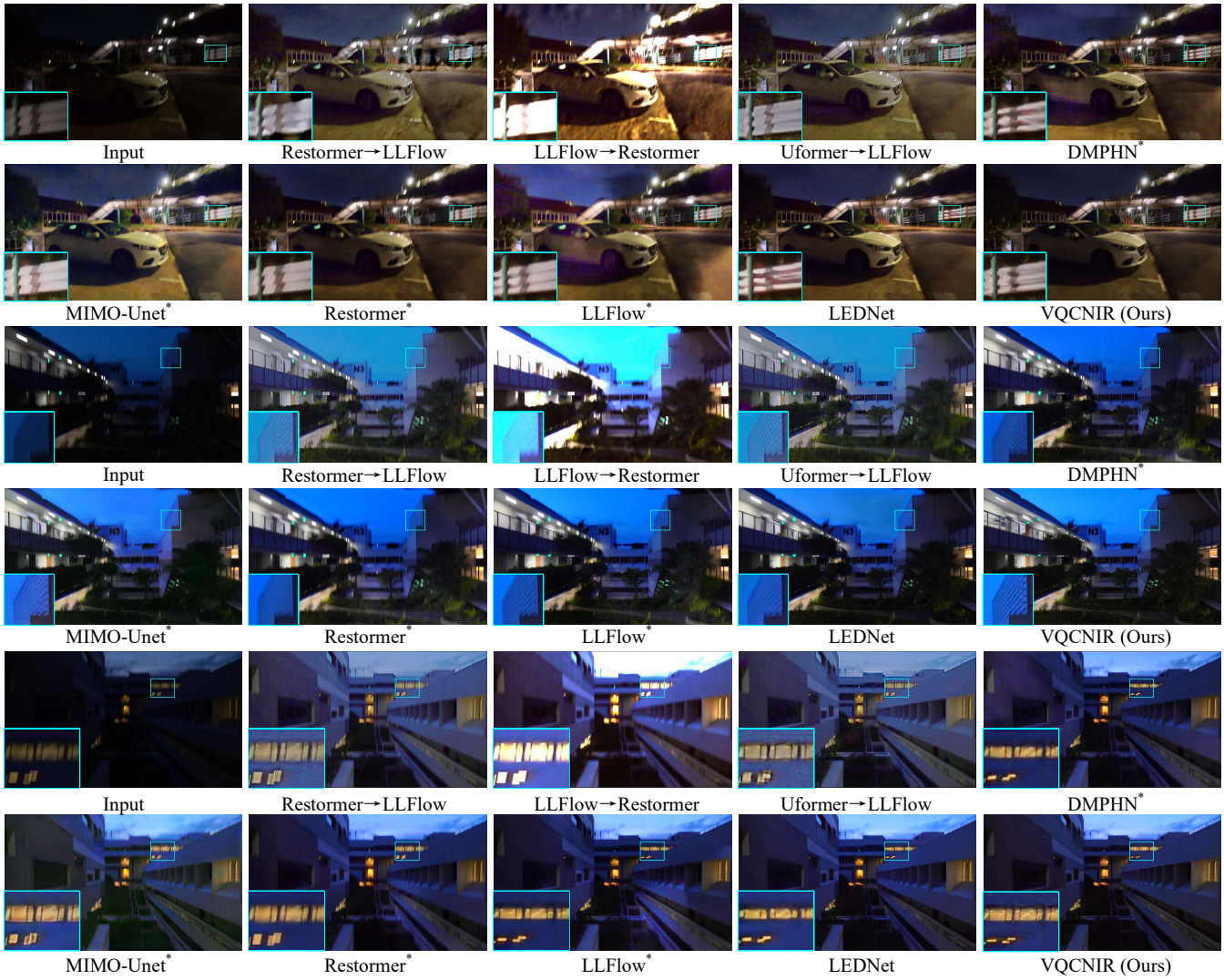

Figure 4: Visual comparison on a real-world night blurry image. The proposed VQCNIR achieves the best perceptual quality with more stable light enhancement and better deblurring performance, especially in saturated regions, while other methods still leave large blurs in saturated regions and suffer from noticeable artifacts. '*' indicates the network is trained with our LOL-Blur dataset (Zhou, Li, and Change Loy 2022). (Zoom in for the best view)

Input LEDNet VQCNIR (Ours)

Input LEDNet VQCNIR (Ours)

Figure 5: Visual results on RealBlur dataset (Rim et al. 2020). The proposed VQCNIR performs well in different scenarios. (Zoom in for the best view)

Input LEDNet VQCNIR (Ours)

Input LEDNet VQCNIR (Ours)

Figure 6: Visual results on RealBlur dataset (Rim et al. 2020). The proposed VQCNIR performs well in different scenarios. (Zoom in for the best view)