# Learnable Feature Patches and Vectors for Boosting Low-light Image Enhancement without External Knowledge

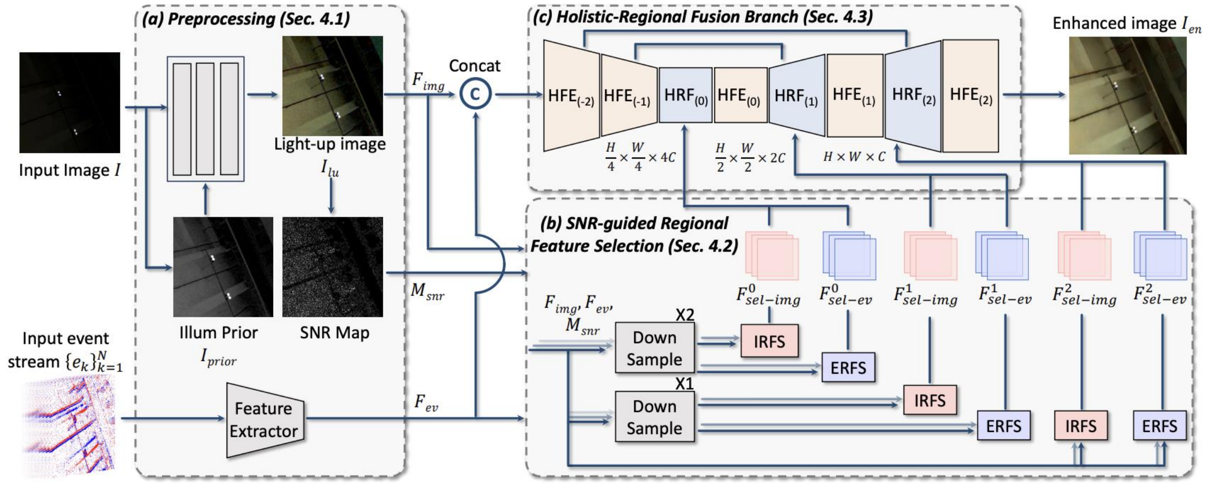
Xiaogang Xu, Jiafei Wu, Qingsen Yan,
Jiequan Cui, Richang Hong, Bei Yu

ICCV HONOLULU HAWAII
OCT 19-23, 2025

# Low-light Image Enhancement

**Motivation:**

- Low-light image Enhancement is a highly ill-posed task.
- Additional information can help the performance, so-called “reference”.
- Data collected from other devices, but it’s not very practical, e.g., [1].

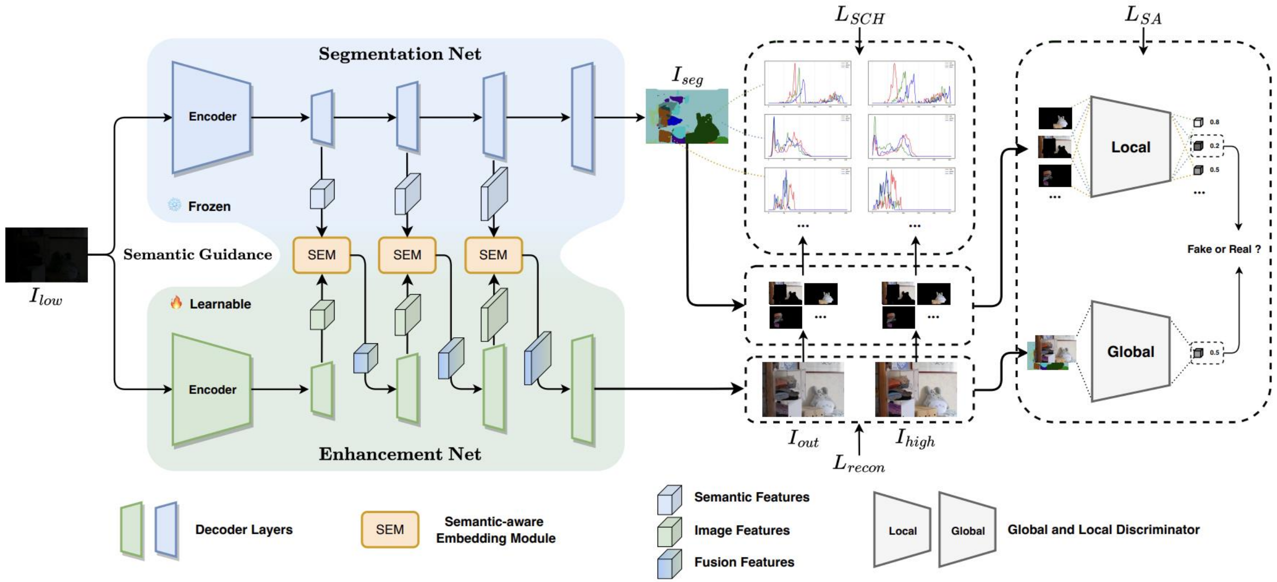



[1] Liang G, Chen K, Li H, et al. Towards robust event-guided low-light image enhancement: a large-scale real-world event-image dataset and novel approach[C] CVPR2024

# Low-light Image Enhancement

**Motivation:**

- Low-light image Enhancement is a highly ill-posed task.
- Additional information can help the performance, so-called “reference”.
- Pre-trained models, e.g., [2].

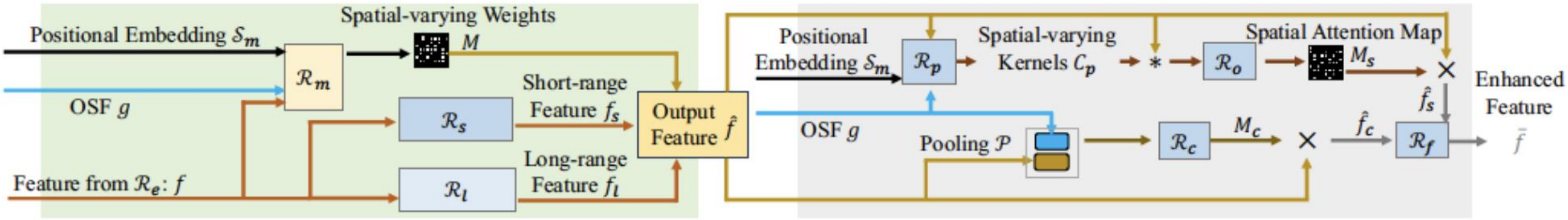



[2] Wu Y, Pan C, Wang G, et al. Learning semantic-aware knowledge guidance for low-light image enhancement[C] CVPR2023.

# Low-light Image Enhancement

**Motivation:**

- Low-light image Enhancement is a highly ill-posed task.
- Additional information can help the performance, so-called “reference”.
- Learnable features, e.g., [3].




(a) Pre-Train-Guided Spatial-Varying Enhancement $\mathcal{A}$ (b) Pre-Train-Guided Channel-Spatial Attention $\mathcal{C}$

[3] Xu X, Kong S, Hu T, et al. Boosting image restoration via priors from pre-trained models[C] CVPR2024

# Low-light Image Enhancement

**Motivation:**

- Low-light image Enhancement is a highly ill-posed task.
- Additional information can help the performance, so-called “reference”.
- *Existing methods overlook the valuable references hidden within the training dataset itself.*

# Learnable Feature Patches and Vectors (LFPVs)

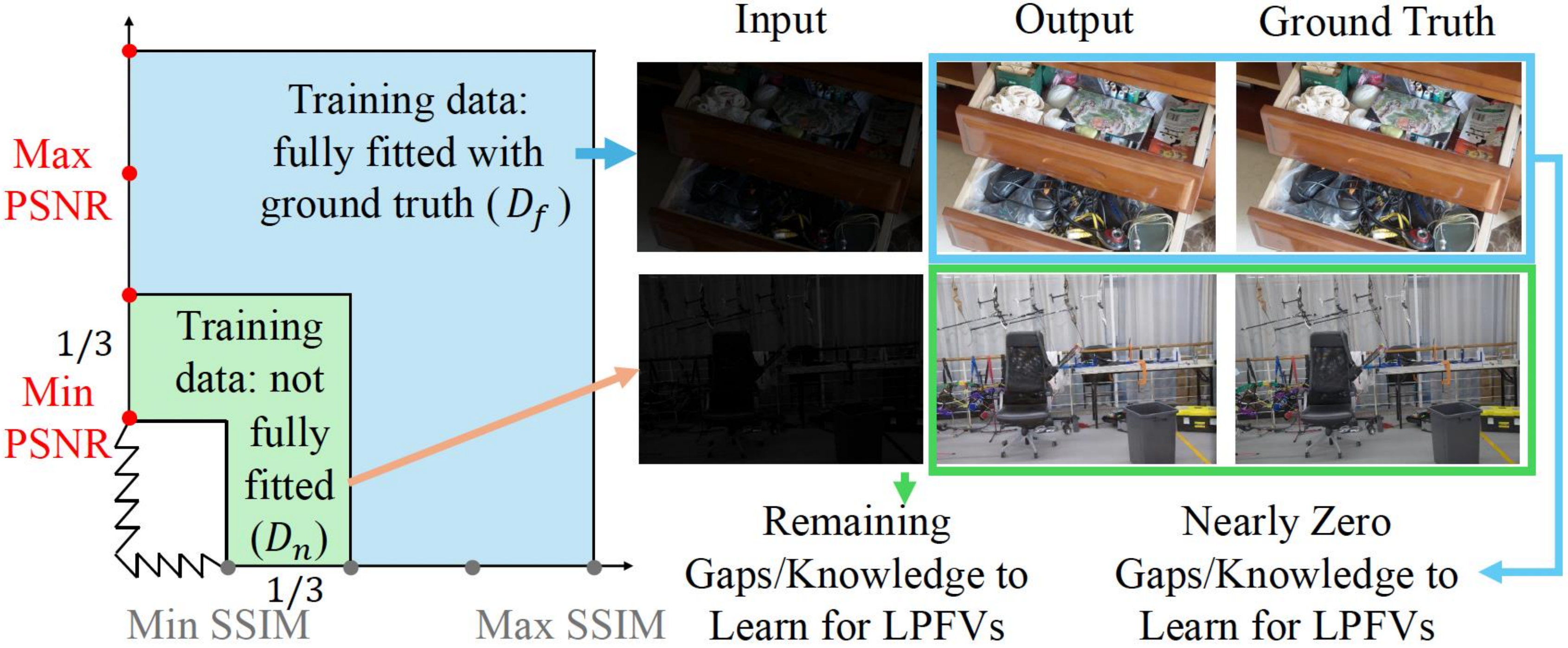
**LFPVs:**

- The training set can be divided into two subsets.
- "images that are fully fitted by the network" $D_f$ :loss approaching zero in the supervised setting.
- "images that are not fully fitted" $D_n$
- The network F has totally learned the mapping relationships for $D_f$ and only partial relationships for $D_n$

# Learnable Feature Patches and Vectors (LFPVs)

**LFPVs:**

- LFPVs mainly capture knowledge from $D_n$ that has not be totally learned by LLIE network.



# Learnable Feature Patches and Vectors (LFPVs)

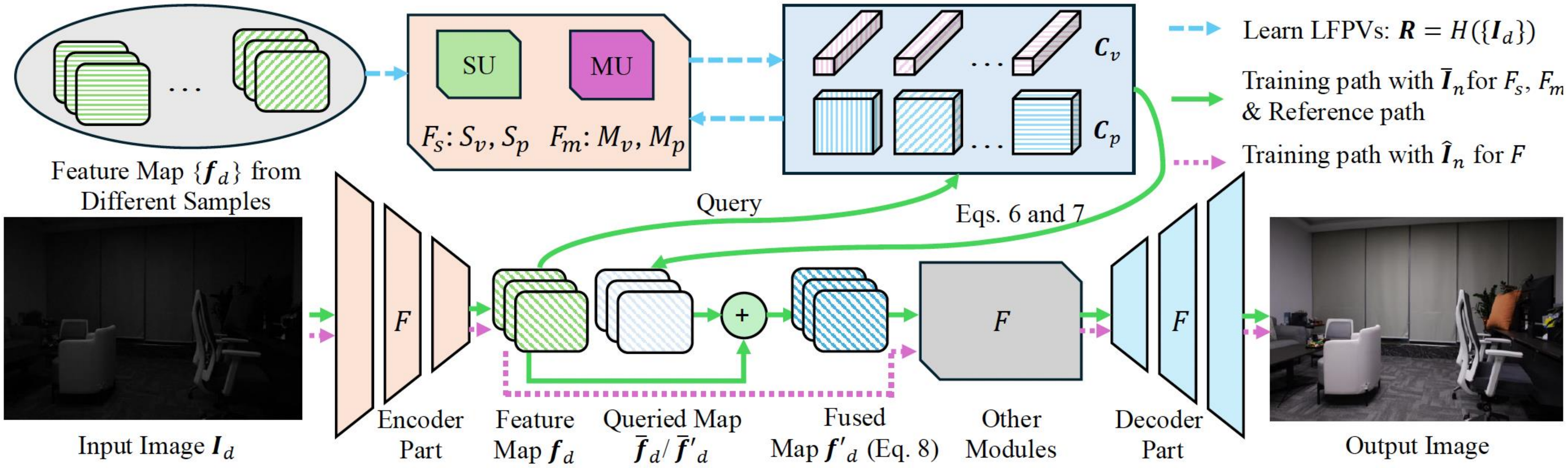
**LFPVs:**

- Different feature vectors or patches in LFPVs are treated as nodes in a graph structure with mutual connections.
- Each node is updated via a Sample Updater (SU, $F_s$), and information can be propagated among LFPVs via a Mutual Updater (MU, $F_m$).
- SU and MU are implemented with additional lightweight networks.

# Learnable Feature Patches and Vectors (LFPVs)

**LFPVs:**

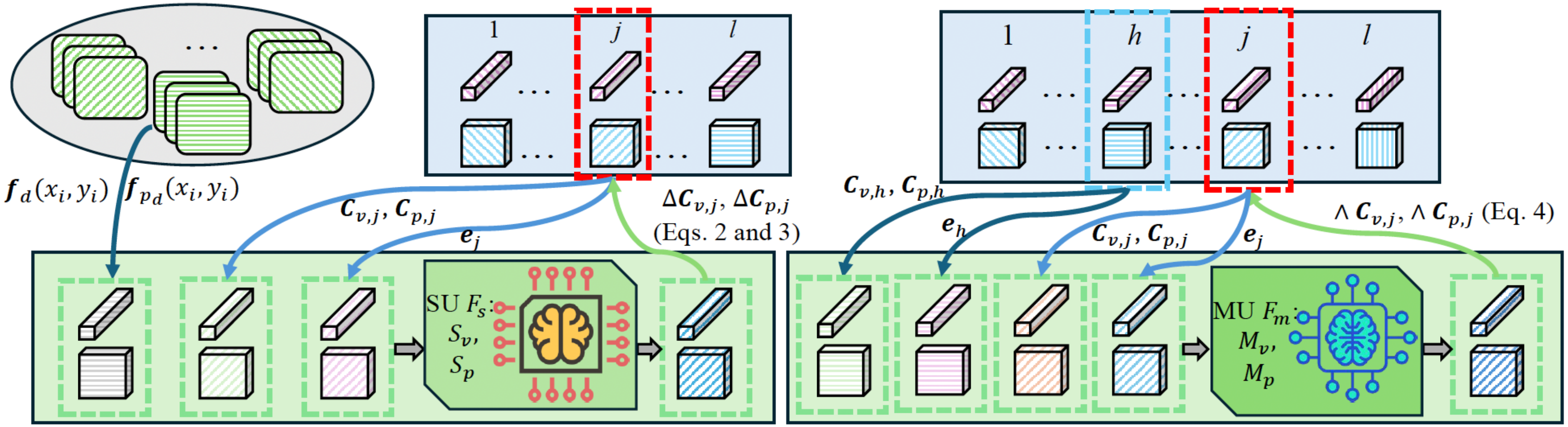
- The learning can be incorporated into existing LLIE frameworks.
- During inference, SU and MU are removed.



# Learnable Feature Patches and Vectors (LFPVs)

**LFPVs:**

- SU will utilize the extracted features from different samples and produce the update for LFPVs with the corresponding identity embedding.
- MU builds the bridge between arbitrary two nodes of LFPVs, mutually propagating their information with LFPVs content and identity embeddings.

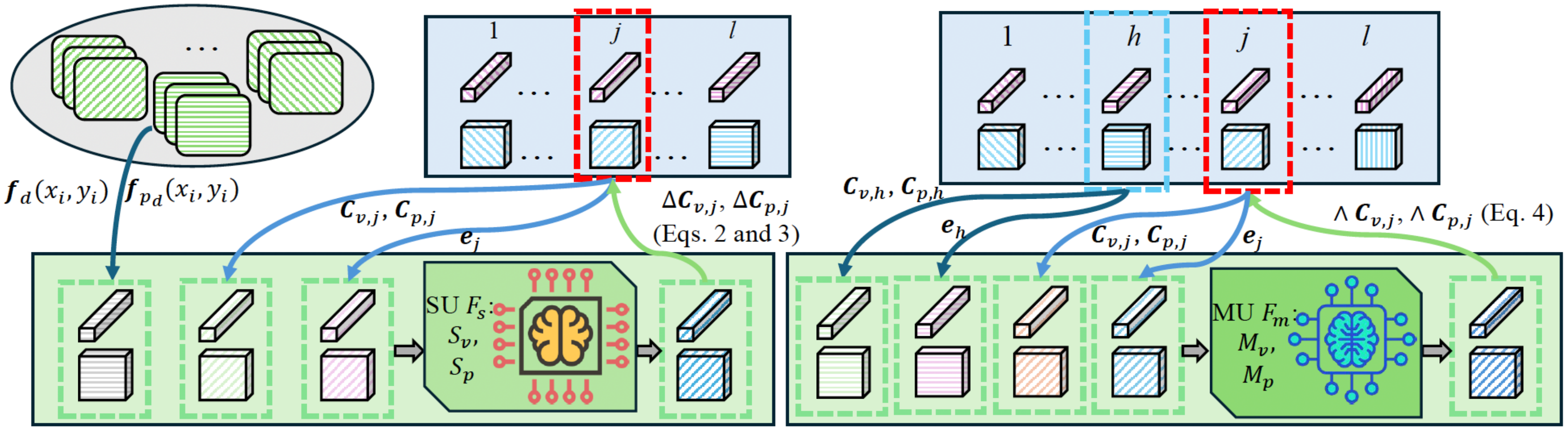


# Learnable Feature Patches and Vectors (LFPVs)

**LFPVs:**

- *Suppose LFPVs are $C_v \in \mathbb{R}^{l\times c}$ for vector and $C_p \in \mathbb{R}^{l\times(c\times k\times k)}$ for patch*
- *SU will utilize the extracted features from different samples and produce the update for LFPVs with the corresponding identity embedding, e.g., $e_j$.*

$$\Delta \boldsymbol{C}_{v,j} = S_v(\boldsymbol{f}_d(x_i, y_i) \oplus \boldsymbol{C}_{v,j} \oplus \boldsymbol{e}_j), j \in [1, l], \quad \Delta \boldsymbol{C}_{p,j} = S_p(\boldsymbol{f}_{p_d}(x_i, y_i) \oplus \boldsymbol{C}_{p,j} \oplus \boldsymbol{e}_j),$$

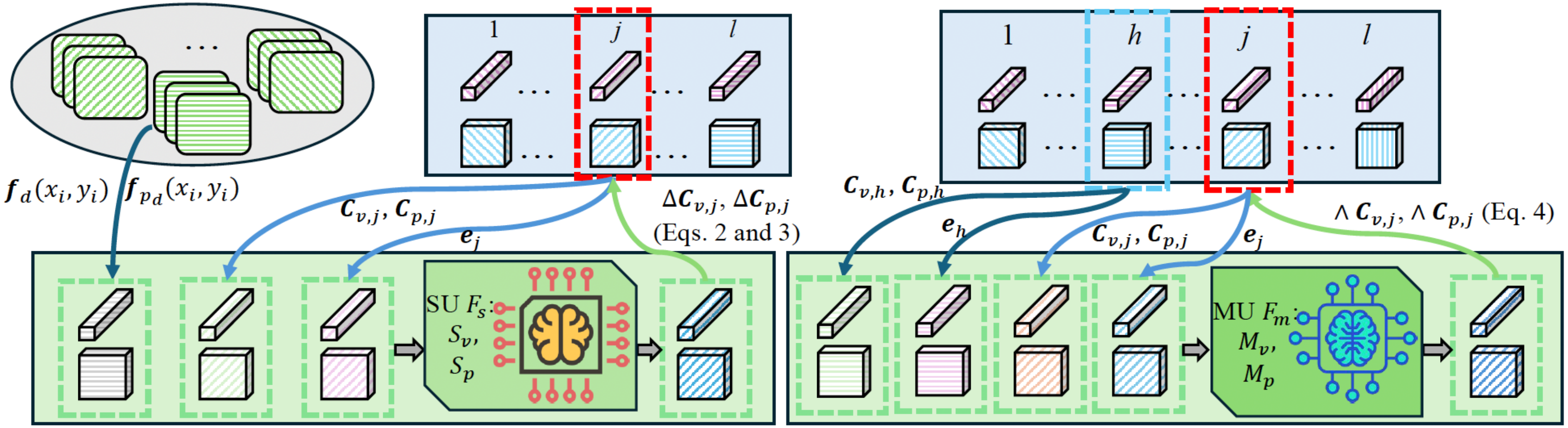


# Learnable Feature Patches and Vectors (LFPVs)

**LFPVs:**

- *Suppose LFPVs are* $C_v \in \mathbb{R}^{l\times c}$ *for vector and* $C_p \in \mathbb{R}^{l\times(c\times k\times k)}$ *for patch*
- *MU builds the bridge between arbitrary two nodes of LFPVs, mutually propagating their information with LFPVs content and identity embeddings.*

$$\wedge \boldsymbol{C}_{v,j} = M_v(\boldsymbol{C}_{v,j} \oplus \boldsymbol{C}_{v,h} \oplus \boldsymbol{e}_j \oplus \boldsymbol{e}_h), \quad \wedge \boldsymbol{C}_{p,j} = M_p(\boldsymbol{C}_{p,j} \oplus \boldsymbol{C}_{p,h} \oplus \boldsymbol{e}_j \oplus \boldsymbol{e}_h),$$



# Learnable Feature Patches and Vectors (LFPVs)

**LFPVs:**

- *Overall update is the sum of SU and MU*

$$\boldsymbol{C}_{v/p,j}^{o+1} = \boldsymbol{C}_{v/p,j}^{o} + \Delta_j^o + \wedge_j^o,$$

$$\Delta_j^o = \sum\nolimits_{\boldsymbol{f}_d/\boldsymbol{f}_{p_d}, x_i, y_i} (\Delta \boldsymbol{C}_{v/p,j}),$$

$$\wedge_j^o = \sum\nolimits_h (\wedge \boldsymbol{C}_{v/p,j}),$$

# Use LFPVs

- *Querying with $C_v$* (Res. and S.M. denote the feature reshape and softmax operations)

$$
\begin{aligned}
&\hat{\boldsymbol{f}}_d = \text{Res.}(\boldsymbol{f}_d) \in \mathbb{R}^{(h\times w)\times c}, \boldsymbol{W}_v = \text{Res.}(\boldsymbol{C}_v) \in \mathbb{R}^{c\times l}, \\
&\boldsymbol{T}_v = \hat{\boldsymbol{f}}_d * \boldsymbol{W}_v \in \mathbb{R}^{(h\times w)\times l}, \hat{\boldsymbol{T}}_v = \text{S.M.}(\boldsymbol{T}_v, \text{dim}=1), \\
&\bar{\boldsymbol{f}}_d = \hat{\boldsymbol{T}}_v * \boldsymbol{C}_v \in \mathbb{R}^{(h\times w)\times c},
\end{aligned}
$$

# Use LFPVs

- *Querying with $C_p$* (Res. and S.M. denote the feature reshape and softmax operations)

$$
\begin{aligned}
&\hat{\boldsymbol{f}}'_{p_d} = \text{Res.}(\boldsymbol{f}_d) \in \mathbb{R}^{(h'\times w')\times c'}, \boldsymbol{W}_p = \text{Res.}(\boldsymbol{C}_p) \in \mathbb{R}^{c'\times l}, \\
&\boldsymbol{T}_p = \hat{\boldsymbol{f}}'_{p_d} * \boldsymbol{W}_p \in \mathbb{R}^{(h'\times w')\times l}, \hat{\boldsymbol{T}}_p = \text{S.M.}(\boldsymbol{T}_p, \text{dim} = 1), \\
&\bar{\boldsymbol{f}}'_d = \hat{\boldsymbol{T}}_p * \boldsymbol{C}_p \in \mathbb{R}^{(h\times w)\times c},
\end{aligned}
\tag{7}
$$

where $h' = h/k$, $w' = w/k$, $c' = c \times k \times k$.

# Use LFPVs

- *The final results are the fusion of three features*

$$\boldsymbol{f}'_d = \bar{\boldsymbol{f}}_d + \bar{\boldsymbol{f}}'_d + \boldsymbol{f}_d,$$

- *The final loss function*

$$\mathcal{L} = \mathcal{L}(\hat{\boldsymbol{I}}_n, \boldsymbol{I}_n) + \lambda \mathcal{L}(\bar{\boldsymbol{I}}_n, \boldsymbol{I}_n) = \|\hat{\boldsymbol{I}}_n - \boldsymbol{I}_n\| + \lambda \|\bar{\boldsymbol{I}}_n - \boldsymbol{I}_n\|,$$

*Encourage F to learn as much knowledge as possible*

*Guide $F_s$ and $F_m$ in formulating* LFPVs to capture the remaining knowledge

# Experiments

- *The enhancement for existing methods*

| | SID | | SMID | | SDSD-Indoor | | SDSD-Outdoor | |
|---|---|---|---|---|---|---|---|---|
| Methods | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM |
| MIRNet [34] | 20.84 | 0.605 | 25.66 | 0.762 | 24.38 | 0.864 | 27.13 | 0.837 |
| MIRNet +Ours | **21.98** | **0.629** | **26.48** | **0.774** | **26.60** | **0.882** | **28.28** | **0.859** |
| SNR [29] | 22.87 | 0.625 | 28.08 | 0.801 | 28.47 | 0.882 | 25.70 | 0.804 |
| SNR+Ours | **23.15** | **0.648** | **28.35** | **0.807** | **29.09** | **0.894** | **26.01** | **0.816** |
| R.M. [35] | 22.27 | 0.649 | 26.97 | 0.758 | 25.67 | 0.827 | 24.79 | 0.802 |
| R.M.+Ours | **23.60** | **0.664** | **28.63** | **0.775** | **27.06** | **0.846** | **26.77** | **0.814** |
| LLFlow [25] | 21.72 | 0.618 | 27.84 | 0.803 | 26.51 | 0.883 | 26.02 | 0.859 |
| LLFlow +Ours | **23.16** | **0.639** | **28.56** | **0.815** | **28.44** | **0.901** | **28.82** | **0.873** |
| R.F. [1] | 24.44 | 0.680 | 29.15 | 0.815 | 29.77 | 0.896 | 29.49 | 0.877 |
| R.F.+Ours | **24.66** | **0.697** | **30.07** | **0.826** | **31.52** | **0.914** | **31.61** | **0.901** |
| Diff-L [6] | 21.45 | 0.571 | 27.57 | 0.783 | 23.93 | 0.836 | 24.19 | 0.832 |
| Diff-L+Ours | **22.30** | **0.603** | **28.72** | **0.808** | **26.38** | **0.859** | **26.86** | **0.857** |
| Event [12] | - | - | - | - | 28.52 | 0.913 | 26.67 | 0.836 |
| Event+Ours | - | - | - | - | **28.87** | **0.925** | **27.36** | **0.847** |

# Experiments

- *The comparison with other types of references*

| Methods | SNR [29] | +SKF [27] | +SMG(sem.) [30] | +SMG(dep.) [30] | +Pretrain [31] | +ACCA [39] | +Ours |
|---|---|---|---|---|---|---|---|
| PSNR ↑ | 21.48 | 23.05 | 24.84 | 24.12 | 25.50 | 24.00 | 24.38 |
| SSIM ↑ | 0.849 | 0.853 | 0.880 | 0.851 | 0.892 | 0.872 | 0.864 |
| +Params | 0 | 2.15M | 16.76M | 16.76M | 0.67M | 0.088M | 0.017M |
| Methods | URetinex [26] | +SKF [27] | +SMG(sem.) [30] | +SMG(dep) [30] | +Pretrain [31] | +ACCA [39] | +Ours |
| PSNR ↑ | 21.16 | 23.51 | 23.74 | 23.25 | 24.70 | 23.67 | 24.15 |
| SSIM ↑ | 0.840 | 0.856 | 0.852 | 0.849 | 0.878 | 0.850 | 0.862 |

Table 3. Quantitative comparisons on the LOL-real dataset for various methods that learn references. "+Params" denotes the number of additional parameters employed during inference.

# Experiments

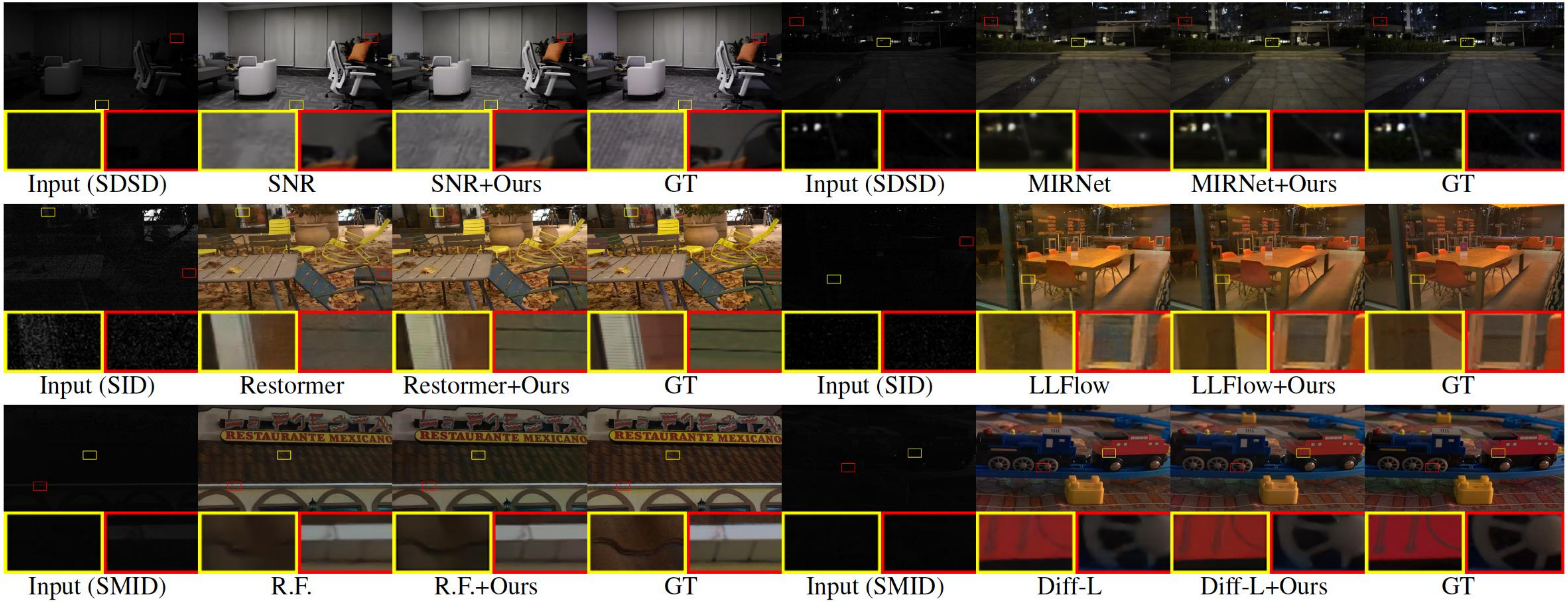
- *The improvement for methods with references*

| Methods | SNR+SKF | +Ours | SNR+SMG | +Ours | SNR+Pretrain | +Ours | SNR+ACCA | +Ours | SNR+CodeBook [14] | +Ours | SNR+LUT [13] | +Ours |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PSNR ↑ | 23.05 | **24.07** | 24.84 | **25.60** | 25.50 | **25.77** | 24.00 | **24.91** | 24.05 | **24.68** | 23.22 | **23.74** |
| SSIM ↑ | 0.853 | **0.865** | 0.880 | **0.891** | 0.892 | **0.906** | 0.872 | **0.886** | 0.860 | **0.871** | 0.858 | **0.874** |
| Methods | UR.+SKF | +Ours | UR.+SMG | +Ours | UR.+Pretrain | +Ours | UR.+ACCA | +Ours | UR.+CodeBook [14] | +Ours | UR.+LUT [13] | +Ours |
| PSNR ↑ | 23.51 | **24.10** | 23.74 | **24.23** | 24.70 | **25.50** | 23.67 | **24.02** | 23.42 | **24.19** | 22.83 | **23.41** |
| SSIM ↑ | 0.856 | **0.862** | 0.852 | **0.868** | 0.878 | **0.890** | 0.850 | **0.852** | 0.854 | **0.863** | 0.847 | **0.854** |

Table 4. Quantitative comparison on the LOL-real dataset, showing that our method can also improve the performance of existing methods learning references. Note that we employ the strategy of learning codebook [14] and lookup table (LUT) [13] into different LLIE architectures for a fair comparison. U.R. denotes URetinex [26].

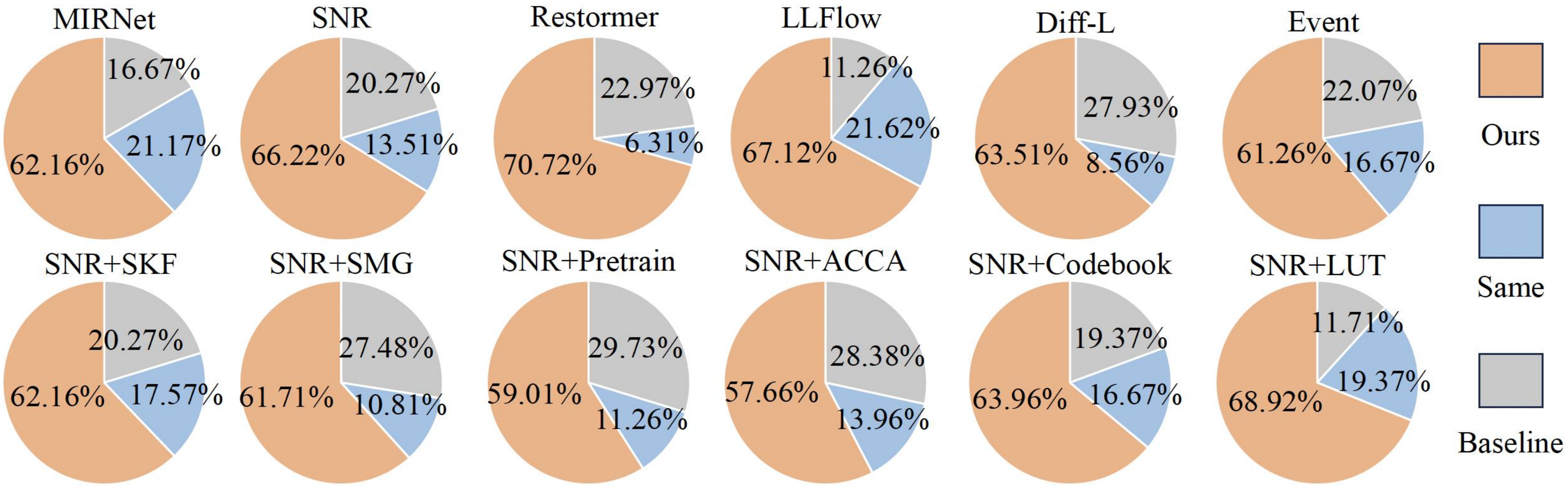
# Experiments

- *Visual comparison*



# Experiments

- *User Study: AB-test, choose "ours" or "baseline" or "the same"*



# Experiments

- *Improvement for downstream tasks*
- *Include nighttime image classification and semantic segmentation*

| Methods | E.GAN [7] | +Ours | LEDNet [41] | +Ours | Z.D.+ [11] | +Ours | RUAS [16] | +Ours | SCI [19] | +Ours | UR. [26] | +Ours |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Top-1 (%) on CODaN ↑ | 56.68 | **58.02** | 57.40 | **58.61** | 57.96 | **59.08** | 58.36 | **59.14** | 58.68 | **59.54** | 58.72 | **59.87** |
| mIoU on Nighttime Driving ↑ | 25.2 | **26.4** | 27.6 | **28.2** | 32.7 | **33.5** | 25.1 | **26.0** | 28.6 | **29.3** | 28.1 | **29.6** |
| mIoU on Dark-Zurich ↑ | 24.9 | **26.0** | 26.6 | **27.8** | 28.3 | **29.1** | 23.4 | **24.7** | 25.7 | **26.5** | 24.0 | **24.8** |

# Thanks