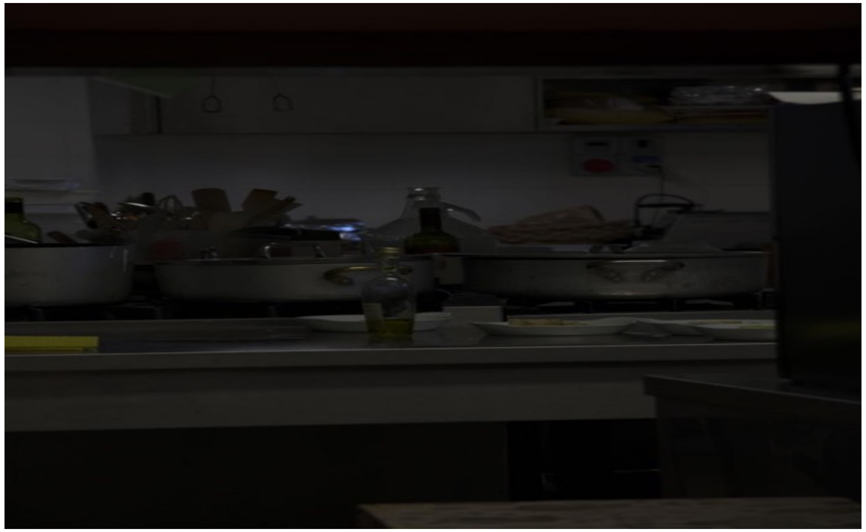
# **CLODE**: Continuous Exposure Learning for Low-light Image Enhancement using Neural ODEs

Donggoo Jung[*1], Daehyun Kim[*1], Tae Hyun Kim[†2]
Dept. of Artificial Intelligence [1], Dept. of Computer Science[2], Hanyang University

# Low-light image enhancement

Images captured in low-light condition

# Low-light image enhancement

Images captured in well exposed condition

# Curve-adjustment method: gamma correction

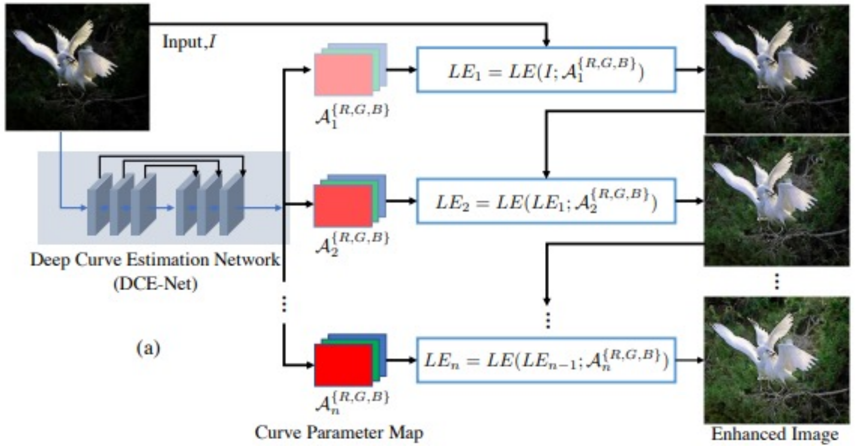
$$I_{out} = I_{in}^{\gamma}$$

$$\gamma = 0.6$$

Gamma correction:

- Overall exposure enhancement
- Cannot consider exposure enhancement locally

# Curve-adjustment method

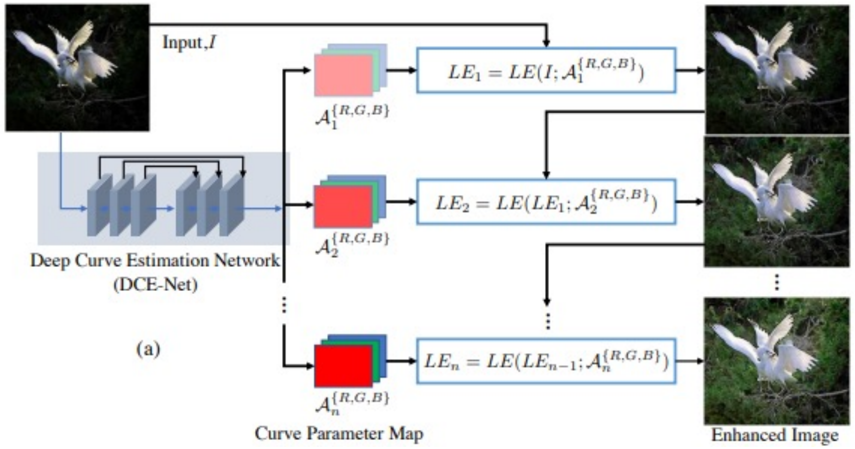



$$LE(x) = LE(x) + \alpha \cdot \big(1 - LE(x)\big) \cdot LE(x)$$

$$0 \leq LE(x), \leq 1,\ -1 \leq \alpha \leq 1$$

- Pixel-wise curve adjustment method
- Iterative method makes higher-order curves for enhancing low-light images
- Detail preservation
- Trained on zero-reference loss functions

# Curve-adjustment method



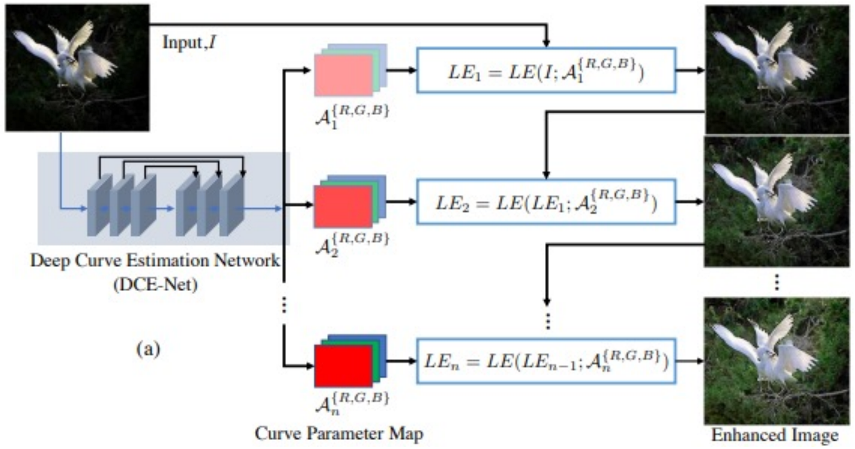

$$LE(x) = LE(x) + \alpha \cdot \big(1 - LE(x)\big) \cdot LE(x)$$

$$0 \leq LE(x), \leq 1, \; -1 \leq \alpha \leq 1$$

- Pixel-wise curve adjustment method
- Iterative method makes higher-order curves for enhancing low-light images
- Detail preservation
- Trained on Zero-reference loss functions

- Update process is in discrete-space
- Cannot guarantee convergence of the optimization

# Curve-adjustment method




$$LE(x) = LE(x) + \mathcal{A}(x) \cdot \big(1 - LE(x)\big) \cdot LE(x)$$

$$0 \leq LE(x), \leq 1, \; -1 \leq \mathcal{A}(x) \leq 1$$

- Pixel-wise curve adjustment method
- Iterative method makes higher-order curves for enhancing low-light images
- Detail preservation
- Trained on Zero-reference loss functions

- Update process is in discrete-space
- Cannot guarantee convergence of the optimization

- Discrete-space update process to **continuous-space update process via Neural ODE**
- Can find input-specific higher-order curves until convergence

# Reformulation

**Iteration formulation:**

- $I_1 = I_0 + \mathcal{A}_0 \otimes I_0 \otimes (1 - I_0)$
- $I_2 = I_1 + \mathcal{A}_1 \otimes I_1 \otimes (1 - I_1)$
- $I_3 = I_2 + \mathcal{A}_2 \otimes I_2 \otimes (1 - I_2)$
- $\dots$
- $I_N = I_{N-1} + \mathcal{A}_{N-1} \otimes I_{N-1} \otimes (1 - I_{N-1})$

$\mathcal{A}$: Cureve parameter map $(-1 \leq \mathcal{A}(x) \leq 1)$
$I$ : Image

# Reformulation

**Iterative formulation:**

- $I_1 = I_0 + \mathcal{A}_0 \otimes I_0 \otimes (1 - I_0)$
- $I_2 = I_1 + \mathcal{A}_1 \otimes I_1 \otimes (1 - I_1)$
- $I_3 = I_2 + \mathcal{A}_2 \otimes I_2 \otimes (1 - I_2)$
- ...
- $I_N = I_{N-1} + \mathcal{A}_{N-1} \otimes I_{N-1} \otimes (1 - I_{N-1})$

$\Rightarrow$

**Ordinary Differential Equation:**

- $I_{n+1} = I_n + \mathcal{A}_n \otimes I_n \otimes (1 - I_n)$ **:** conventional formula
- $I_{t+1} = I_t + f_\theta(I_t, t)$ **:** continuous variable $\boldsymbol{t}$
- $\frac{dI_t}{dt} = f_\theta(I_t, t)$

# Reformulation

**Iterative formulation:**

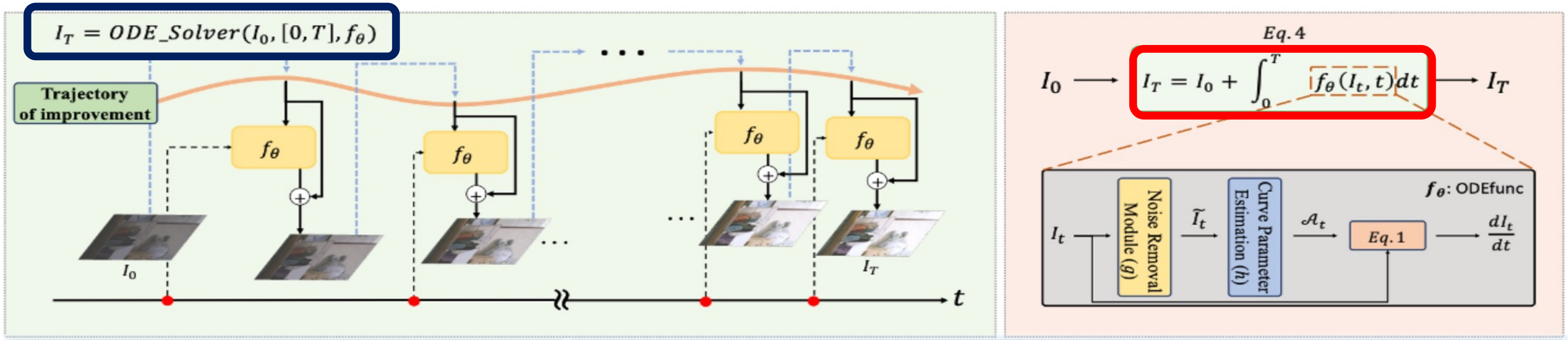
- $I_1 = I_0 + \mathcal{A}_0 \otimes I_0 \otimes (1 - I_0)$
- $I_2 = I_1 + \mathcal{A}_1 \otimes I_1 \otimes (1 - I_1)$
- $I_3 = I_2 + \mathcal{A}_2 \otimes I_2 \otimes (1 - I_2)$
- $\dots$
- $I_N = I_{N-1} + \mathcal{A}_{N-1} \otimes I_{N-1} \otimes (1 - I_{N-1})$

**Ordinary Differential Equation:**

- $I_{n+1} = I_n + \mathcal{A}_n \otimes I_n \otimes (1 - I_n)$ **:** conventional formula
- $I_{t+1} = I_t + f_\theta(I_t, t)$ **:** continuous variable $\boldsymbol{t}$
- $\frac{dI_t}{dt} = f_\theta(I_t, t)$
- $\boxed{I_T = I_0 + \int_o^T f_\theta(I_t, t)dt}$
- $\boxed{I_T = ODE\_Solver(I_0, [0, T], f_\theta)}$

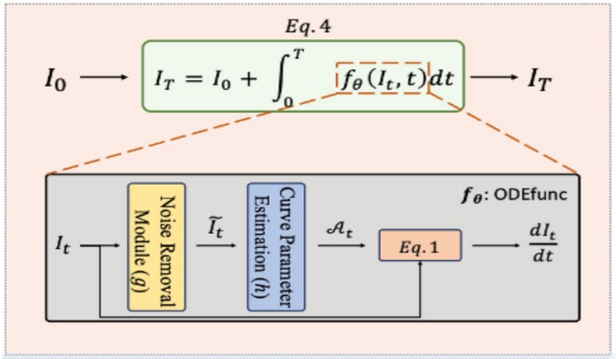
* ODE solver: Euler, RK4, dopri5, …

# Overall




$$(Eq.1)\quad I_{n+1} = I_n + \mathcal{A}_n \otimes I_n \otimes (1 - I_n)$$

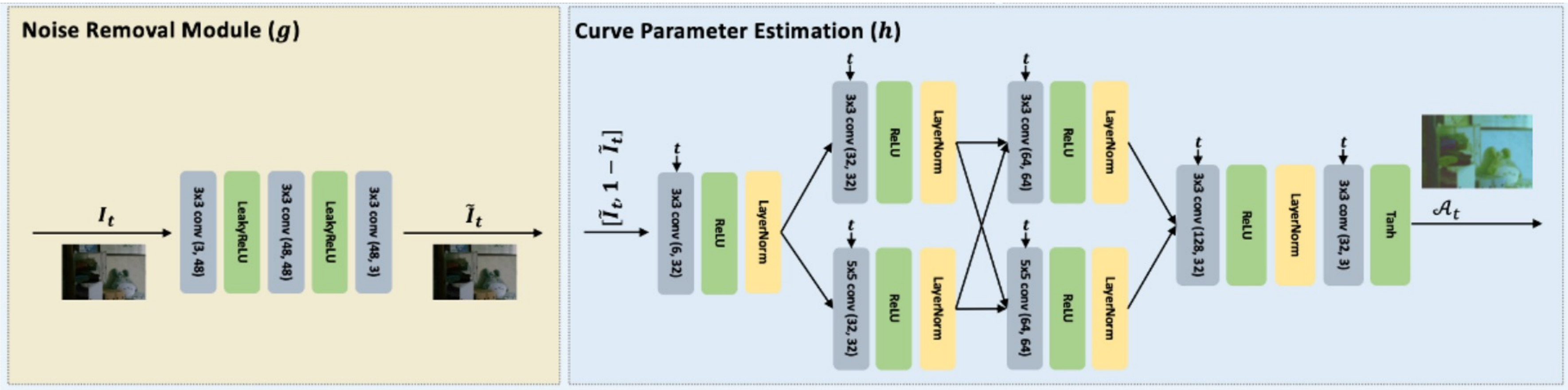
# Network overall



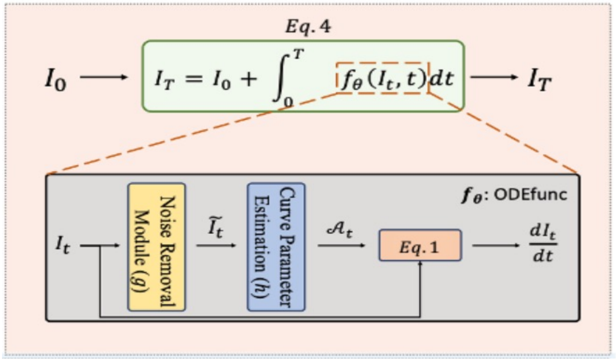

- Noise Removal ($g$):
  Provide clean input images for Curve Parameter Estimation module

- Curve Parameter Estimation ($h$):
  Estimate curve parameter map of each state

$$(Eq. 1) \quad I_{n+1} = I_n + \mathcal{A}_n \otimes I_n \otimes (1 - I_n)$$

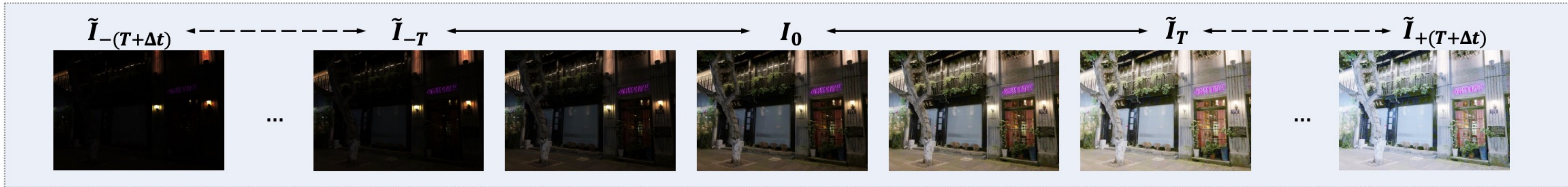


# Inference




- Inference process:
  After getting $I_T$, apply Noise Removal Module ($g$) for noise free output $\tilde{I}_T$ ($= g(I_T)$)

- User controllable design:
  By changing $T$, the exposure level of the output image can be easily controlled.



# Loss functions

- Spatial Consistency Loss: $$\mathcal{L}_{spa} = \frac{1}{K}\sum_{i=1}^{K}\sum_{j\in\Omega(i)}(|m_4(I_T)_i - m_4(I_T)_j| - |m_4(I_0)_i - m_4(I_0)_j|)^2.$$
- Color Constancy Loss: $$\mathcal{L}_{col} = (R-B)^2 + (R-G)^2 + (G-B)^2$$
- Parameter Regularization Loss: $$\mathcal{L}_{param} = \sum_t(|\nabla_x\mathcal{A}_t| + |\nabla_y\mathcal{A}_t|)^2$$
- Exposure Loss: $$\mathcal{L}_{exp} = ||m_{16}(I_T) - \mathrm{E}||_2^2. \quad (\mathrm{E} = 0.6)$$
- Noise Removal Loss:
  Zero shot noise removal loss function from "Zero-Shot Noise2Noise: Efficient Image Denoising without any Data (CVPR2023)" is applied on Noise Removal Module

# Results

| Training | Method | #Params (M) | Train dataset | LSRW | | | | LOL | | | | Average | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Normal | | *GT Mean* | | Normal | | *GT Mean* | | Normal | | *GT Mean* | |
| | | | | PSNR↑ | SSIM↑ | PSNR↑ | SSIM↑ | PSNR↑ | SSIM↑ | PSNR↑ | SSIM↑ | PSNR↑ | SSIM↑ | PSNR↑ | SSIM↑ |
| Supervised | RetinexNet (Chen Wei, 2018) | 0.4446 | LOL | 15.49 | 0.355 | 16.55 | 0.371 | 16.77 | 0.419 | 17.65 | 0.648 | 16.13 | 0.387 | 17.10 | 0.510 |
| | URetinexNet (Wu et al., 2022a) | 0.3069 | LOL, SICE | 17.63 | 0.516 | 18.10 | 0.523 | 19.84 | 0.826 | 21.33 | 0.835 | 18.74 | 0.671 | 19.71 | 0.679 |
| | DRBN (Yang et al., 2021) | 0.5556 | LOL | 16.15 | 0.542 | 17.68 | 0.548 | 16.29 | 0.617 | 19.55 | 0.746 | 16.22 | 0.580 | 18.62 | 0.647 |
| | KinD (Zhang et al., 2019) | 8.0160 | LOL | 16.47 | 0.493 | 19.86 | 0.504 | 17.65 | 0.775 | 20.87 | 0.802 | 17.06 | 0.634 | 20.36 | 0.653 |
| | LLFlow (Wang et al., 2022b) | 38.859 | LOL | 17.52 | 0.509 | 18.68 | 0.518 | 21.15 | 0.854 | 24.99 | 0.871 | 19.34 | 0.681 | 21.84 | 0.694 |
| | RetinexFormer (Cai et al., 2023) | 1.6057 | LOL | 17.76 | 0.517 | 19.15 | 0.529 | 25.15 | 0.845 | 27.18 | 0.850 | 21.45 | 0.681 | 23.17 | 0.690 |
| Unsupervised | SCI-easy (Ma et al., 2022) | 0.0003 | MIT-5K | 11.79 | 0.317 | 16.97 | 0.426 | 9.58 | 0.369 | 18.55 | 0.501 | 10.69 | 0.343 | 17.76 | 0.464 |
| | SCI-medium (Ma et al., 2022) | 0.0003 | LOL, LSRW | 15.24 | 0.424 | 17.84 | 0.439 | 14.78 | 0.521 | 19.11 | 0.504 | 15.01 | 0.473 | 18.47 | 0.472 |
| | SCI-difficult (Ma et al., 2022) | 0.0003 | DARKFace | 15.16 | 0.408 | 18.04 | 0.424 | 13.81 | 0.526 | 19.64 | 0.510 | 14.48 | 0.467 | 18.84 | 0.467 |
| | SCI* (Ma et al., 2022) | 0.0003 | LOL | 14.82 | 0.413 | 17.65 | 0.437 | 13.84 | 0.507 | 19.02 | 0.499 | 14.33 | 0.460 | 18.34 | 0.468 |
| | RUAS (Liu et al., 2021) | 0.0034 | LOL | 14.27 | 0.470 | 17.10 | 0.509 | 16.41 | 0.500 | 18.65 | 0.520 | 15.34 | 0.485 | 17.88 | 0.514 |
| | ZeroDCE* (Guo et al., 2020) | 0.0794 | LOL | 14.50 | 0.403 | 18.87 | 0.467 | 16.49 | 0.522 | 20.99 | 0.596 | 15.50 | 0.463 | 19.93 | 0.532 |
| | ReLLIE (Zhang et al., 2021b) | - | LOL | - | - | - | - | 18.37 | 0.641 | - | - | - | - | - | - |
| | PairLIE (Fu et al., 2023) | 0.3417 | LOL, SICE | **16.97** | **0.498** | 18.82 | 0.523 | 19.51 | **0.736** | 23.10 | **0.752** | **18.24** | 0.617 | 20.96 | 0.637 |
| | Night-Enhancement (Jin et al., 2022) | 67.011 | LOL | 14.24 | 0.472 | **19.19** | **0.554** | **21.52** | **0.763** | **24.25** | **0.781** | 17.88 | **0.618** | **21.72** | **0.668** |
| | **CLODE** | 0.2167 | LOL | **17.28** | **0.533** | **20.60** | **0.557** | **19.61** | 0.718 | **23.16** | **0.752** | **18.44** | **0.625** | **21.88** | 0.655 |
| | **CLODE** † | 0.2167 | LOL | **20.77** | **0.562** | **20.94** | **0.568** | **23.58** | **0.754** | **24.47** | **0.759** | **22.18** | **0.658** | **22.71** | **0.664** |

**Input** **SCI [11]** **ZeroDCE [6]** **PairLIE [13]** **Night-Enhance [17]** **LLFlow [1]** **RetinexFormer [2]** **CLODE (Ours)** **Ground-Truth**

# Results

Table 2: **Quantitative results on SICE (Cai et al., 2018) Part2**. For a fair comparison, we re-trained some models on SICE Part 1 and marked them with *. Within the unsupervised approaches, the best score is displayed in **red**, the second in **blue** and the third in **black**.

| Training | Method | Train dataset | Normal PSNR↑ | SSIM↑ | LPIPS↓ | NIQE↓ | BRISQUE↓ | PI↓ | Entropy↑ | *GT Mean* PSNR↑ | SSIM↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Supervised | URetinexNet (Wu et al., 2022a) | LOL, SICE | 12.15 | 0.708 | 0.393 | 4.250 | 15.633 | 3.372 | 6.926 | 17.81 | 0.686 |
| | LLFlow* (Wang et al., 2022b) | SICE | 14.34 | 0.608 | 0.279 | 3.643 | 17.011 | 3.481 | 6.566 | 19.59 | 0.658 |
| | ECLNet (Huang et al., 2022b) | SICE | 13.99 | 0.562 | 0.290 | 4.279 | 24.570 | 3.520 | 6.919 | 16.66 | 0.690 |
| | FECNet (Huang et al., 2022a) | SICE | 14.25 | 0.600 | 0.291 | 3.786 | 17.454 | 3.025 | 7.035 | 16.47 | 0.639 |
| | RetinexFormer* (Cai et al., 2023) | SICE | 19.12 | 0.570 | 0.369 | 4.452 | 24.768 | 4.573 | 7.025 | 20.97 | 0.578 |
| | RetinexFormer (Cai et al., 2023) | MIT-5K | 13.23 | 0.564 | 0.263 | 3.848 | 17.350 | 2.863 | 6.881 | 16.35 | 0.609 |
| Unsupervised | SCI-easy (Ma et al., 2022) | MIT-5K | 9.87 | 0.486 | 0.372 | 4.276 | 21.850 | 3.226 | 6.113 | 16.44 | 0.622 |
| | SCI-medium (Ma et al., 2022) | LOL, LSRW | 9.77 | 0.510 | 0.454 | 5.727 | 33.200 | 4.392 | 5.212 | 15.83 | 0.574 |
| | SCI-difficult (Ma et al., 2022) | DarkFace | 11.13 | 0.577 | 0.324 | 4.636 | 23.620 | 3.107 | 6.386 | 16.85 | 0.647 |
| | SCI* (Ma et al., 2022) | SICE | 10.67 | 0.478 | 0.331 | 4.289 | 23.449 | 3.570 | 6.213 | 17.99 | 0.675 |
| | RUAS* (Liu et al., 2021) | SICE | 9.12 | 0.408 | 0.539 | 8.097 | 52.923 | 6.004 | 5.101 | 15.52 | 0.531 |
| | ZeroDCE (Guo et al., 2020) | SICE | 12.67 | **0.635** | **0.244** | **3.886** | **21.630** | **2.821** | 6.516 | 18.85 | **0.686** |
| | PairLIE (Fu et al., 2023) | LOL, SICE | **13.39** | 0.619 | 0.305 | 5.268 | 36.536 | 3.548 | 6.376 | 19.22 | 0.663 |
| | Night-Enhancement* (Jin et al., 2022) | SICE | 13.18 | 0.581 | 0.360 | 4.728 | 33.883 | 4.133 | **6.661** | **19.43** | 0.660 |
| | **CLODE** | SICE | **15.01** | **0.687** | **0.239** | **4.050** | **18.663** | **3.005** | **7.006** | **19.64** | **0.706** |
| | **CLODE†** | SICE | **16.18** | **0.707** | **0.200** | **4.026** | **18.210** | **2.970** | **7.045** | **21.55** | **0.813** |

Table 3: Quantitative comparisons on LOL (Chen Wei, 2018)/SICE (Cai et al., 2018) dataset.

| Method | NIQE↓ | BRISQUE↓ | color-matching histogram loss↓ | $\Delta E_{2000}$↓ (Sharma et al., 2005) | $\Delta E_{ab}$↓ (Sharma & Bala, 2017) |
|---|---|---|---|---|---|
| SCI-easy (Ma et al., 2022) | 7.15/4.28 | 12.42/21.85 | 0.4860/0.4788 | 31.49/27.00 | 39.21/35.13 |
| SCI-medium (Ma et al., 2022) | 7.86/5.73 | 25.87/33.20 | 0.4530/0.4911 | 19.40/27.28 | 27.28/35.96 |
| SCI-difficult (Ma et al., 2022) | 8.06/4.64 | 26.82/23.62 | **0.3854**/0.4872 | 21.06/24.02 | 26.09/31.05 |
| RUAS (Liu et al., 2021) | 6.30/8.10 | 11.98/52.92 | 0.4471/0.5100 | 16.80/29.18 | 29.18/38.83 |
| Zero-DCE (Guo et al., 2020) | 7.78/**3.89** | 27.30/21.63 | 0.4485/0.4647 | 21.93/21.26 | 26.60/27.10 |
| CLODE | **4.52**/4.05 | **8.22**/**18.66** | 0.4381/**0.4606** | **12.73**/**17.04** | **15.71**/**22.39** |
| CLODE† | **4.25**/**4.03** | **8.81**/**18.21** | **0.3848**/**0.4462** | **9.21**/**14.46** | **11.86**/**19.32** |

# Results

## Discrete to Continuous

Table 4: Comparative experiments according to using NODE on LSRW (Hai et al., 2023)/LOL (Chen Wei, 2018).The "Discrete" refers to performing curve adjustment in discrete steps, similar to the conventional methods (Guo et al., 2020; Zhang et al., 2021b), and "Continuous" refers to the reformulation of NODE.

| Method | Case | Step ($N$) | PSNR↑ | SSIM↑ | BRISQUE↓ |
|---|---|---|---|---|---|
| Discrete | (a1) | 1 | 11.19/9.236 | 0.297/0.362 | 41.137/41.169 |
| | (b1) | 5 | 16.12/17.47 | 0.419/0.716 | 31.421/33.042 |
| | (c1) | 10 | 13.94/16.18 | 0.395/0.520 | 32.267/32.243 |
| | (d1) | 20 | 12.95/14.94 | 0.373/0.506 | 33.537/34.941 |
| | (e1) | 30 | 12.87/14.97 | 0.375/0.509 | 33.537/35.342 |
| Continuous | (f1) | $\leq$ 30 (adaptive) | **17.28/19.61** | **0.533/0.718** | **18.426/8.220** |

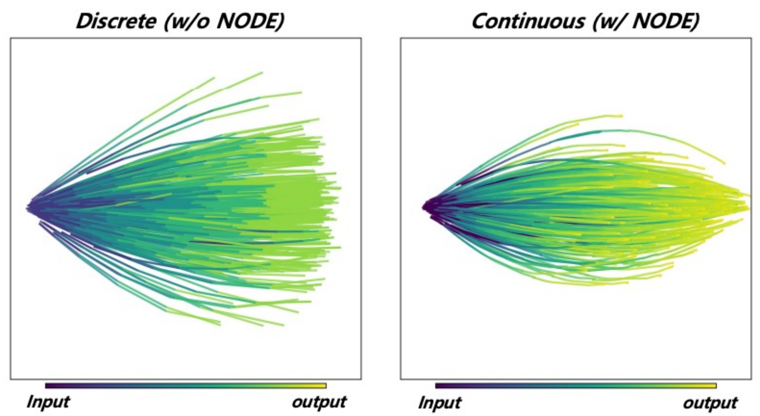
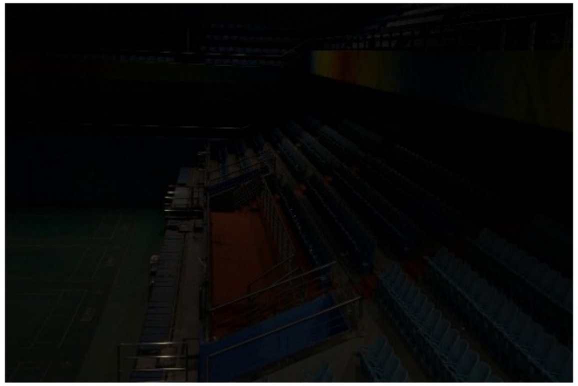



Figure 5: Trajectories of improvement for (e1) and (f1) in Table 4. PCA dimension reduction is used to visualize the trajectories.

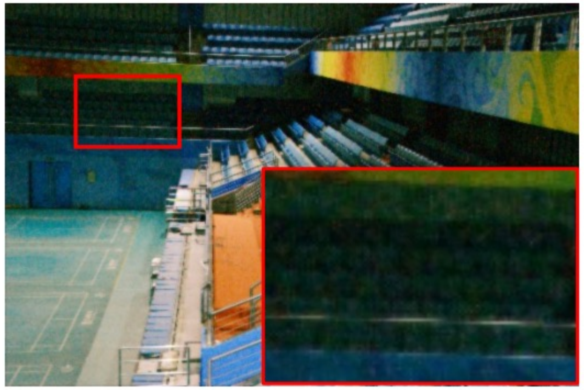
# Results

## Effectiveness of Noise Removal Module



**Input**

**(e3) w/o $\mathcal{L}_{noise}$**



**(f3) CLODE**

# Summary

- **Contribution 1**: CLODE is the first approach to formulate the higher-order curve estimation problem as a Neural ODE problem, enabling effective and accurate solutions with standard ODE solvers.
- **Contribution 2**: By transforming the discrete update formula into Neural ODE, which is solvable in continuous space, we significantly enhance the unsupervised low-light image enhancement results.
- **Contribution 3**: CLODE also offers user controllability without altering the network architecture, enabling users to manually adjust the desired level of exposure as needed.

GitHub https://github.com/dgjung0220/CLODE

{dgjung, daehyun, taehyunkim}@hanyang.ac.kr

https://openreview.net/forum?id=Mn2qgIcIPS