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

Anonymous Author(s)
Affiliation
Address
email

Abstract

Low-light image enhancement poses a significant challenge due to the limited information captured by image sensors in low-light environments. Despite recent improvements in deep learning models, the lack of paired training datasets remains a significant obstacle. Therefore, unsupervised methods have emerged as a promising solution. In this work, we focus on the strength of curve-adjustment-based approaches to tackle unsupervised methods. The majority of existing unsupervised curve-adjustment approaches iteratively estimate higher order curve parameters to enhance the exposure of images while efficiently preserving the details of the images. However, the convergence of the enhancement procedure cannot be guaranteed, leading to sensitivity to the number of iterations and limited performance. To address this problem, we consider the iterative curve-adjustment update process as a dynamic system and formulate it as a Neural Ordinary Differential Equations (NODE) for the first time, and this allows us to learn a continuous dynamics of the latent image. The strategy of utilizing NODE to leverage continuous dynamics in iterative methods enhances unsupervised learning and aids in achieving better convergence compared to discrete-space approaches. Consequently, we achieve state-of-the-art performance in unsupervised low-light image enhancement across various benchmark datasets.

1 Introduction

Images taken in various low-light environments suffer from insufficient light, leading to the capture of limited information by the camera's image sensor. Therefore, many studies have been conducted to improve the quality of the low-light images and achieve images with optimal exposure levels. In particular, recent supervision-based deep learning approaches [1, 2, 3] have shown remarkable performance in enhancing low-light images. However, the process of collecting pairs of low-light scenes and their corresponding ground-truth images for supervised learning is time consuming and resource intensive. As a result, unsupervised approaches that rely solely on low-light images have been proposed to address this problem.

Among many unsupervised low-light image enhancement approaches, curve-adjustment-based methods, conventionally used in photo editing software (*e.g.*, Photoshop), have received much attention. After the introduction of first learning-based curve-adjustment work by Yuan and Sun [7], iterative curve-adjustment-based methods have been explored in various subsequent studies. These unsupervised methods achieve enhancement without using the ground-truth images by fitting the brightness values of pixels in the input image to specific curves. In addition, it is advantageous to preserve local structural information adaptively by allowing efficient pixel-by-pixel computations. For example, ZeroDCE [6, 8] introduced a fast and lightweight neural network to predict pixel-wise curve parameter maps within a fixed iteration step. In addition, ReLLIE [9] produced more accurate

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

(a) (b)

Figure 1: **(a)** Quantitative Evaluation: The average PSNR values on the LSRW [4] and LOL [5], together with the respective parameter numbers for each model. **(b)** Visual Comparisons with ZeroDCE [6] (*unsupervised*), RetinexFormer [2] (*supervised*) and proposed CLODE (*unsupervised*).

image enhancement results by using reinforcement learning to predict the curve parameter map at each iteration step, with users able to adjust the number of iterations.

In general, these curve-adjustment-based methods, which have fewer parameters, offer the advantage of fast and efficient training and also demonstrate the effectiveness of using higher-order curves for low-light image adjustment. However, conventional iterative approaches in discrete-space with fixed update steps do not arrive at the optimal solution and cannot guarantee convergence of the optimization. Therefore, we alleviate this problem in the discrete-space updating process of existing methods. In doing so, we bring out the strengths of curve fitting methods by reformulating the iterative update formula into ordinary differential equations, which allows the iterative approach to be transformed from discrete-space to continuous-space and find input-specific higher-order curves until convergence within a specified tolerance. To be specific, we present the Neural Ordinary Differential Equations (NODE) model for the low-light enhancement task for the first time. By solving the NODE problem using conventional ODE solvers, we obtain better approximate solutions to the curve-adjustment problem, producing more accurate results than conventional results from iterative updates in discrete-space by exploring the continuous exposure dynamics. In this work, we introduce **C**ontinuous exposure learning for **L**ow-light image enhancement using neural **O**rdinary **D**ifferential **E**quations (**CLODE**), which is the first dynamic system for low-light image enhancement. Our main contributions can be summarized as follows:

- CLODE is the first approach to formulate the higher-order curve estimation problem as a NODE problem, enabling effective and accurate solutions with standard ODE solvers.
- By transforming the discrete update formula into NODE, which is solvable in continuous-space, we significantly enhance the unsupervised low-light image enhancement results across various benchmark datasets as shown in Fig. 1. This effectively bridges the performance gap between supervised and unsupervised approaches.
- CLODE also offers user controllability without altering the network architecture, enabling users to manually adjust the desired level of exposure as needed.

2 Related works

2.1 Unsupervised Low-light Image Enhancement

Obtaining well-exposed ground-truth images paired with corresponding low-light images is inherently challenging, which limits the use of supervised learning in low-light image enhancement. To address this limitation, many unsupervised methods have been developed to tackle the problem. First, there are some approaches [10, 11, 12, 13] that utilize the principles of retinex-theory. Among them, PairLIE [13] utilizes retinex-theory to identify the reflectance and illumination, and employs gamma correction with user-defined gamma values to enhance the illumination. In addition, UDCN [14] and HEP [15] use histogram equalization results as a reference for exposure enhancement. Moreover, recent approaches using GANs have shown remarkable improvements by additionally utilizing unpaired images of normal exposed [16, 17]. Lastly, there are curve-adjustment-based methods [6, 8, 18, 9] that transform images through tone mapping. These methods have advanced the curve-fitting techniques from traditional editing tools into deep learning-based approaches, enhancing images by predicting the fitting curves pixel-by-pixel. By repeating the pixel-wise curve fitting and exposure enhancement for a fixed number of iterations in discrete-space, these approaches aim to handle locally varying exposure levels (*i.e.*, single image with both underexposed and overexposed areas) in an unsupervised manner. Our CLODE also follows this unsupervised curve-adjustment-based method and reformulates the curve-fitting problem into a neural ordinary differential equation (NODE). By solving the NODE problem using conventional ODE solvers, we increase the accuracy of curve fitting and thus significantly improve the performance of low-light image enhancement.

2.2 Neural Ordinary Differential Equations

An ordinary differential equation (ODE) is a fundamental concept in mathematics that describes how a function changes with respect to a single variable. It captures the relationship between a function and its derivatives, providing a powerful tool for modeling dynamic systems, such as Newton's Second Law of Motion. To effectively apply the strength of ordinary differential equations to the deep learning model, the concept of neural ordinary differential equations (NODE) is introduced in [19]. The use of NODE facilitates model definition and evaluation, highlighting its effectiveness in parameter efficiency, adaptive computation, and modeling continuous data. In order to effectively capture more complicated functions, the Augmented Neural ODE (ANODE) [20] has been introduced. Furthermore, for seamless continuous time-series modeling, Latent ODE [21] is proposed and recently, ClimODE [22] proposed a continuous-time NODE models for numerical weather prediction. To be specific, in the field of computer vision, the Vid-ODE approach [23] has been introduced to generate continuous-time videos. NODEO [24] has presented a versatile architecture tailored for deformable image registration, and a temporal deformation model using the capabilities of NODE has been developed in [25] to address the challenges associated with future prediction tasks in the context of 4D reconstruction. With advantages like continuous-space modeling, adaptive computation, and memory efficiency, NODE [19] is utilized in various deep learning tasks. However, it has not been extensively explored in the field of image restoration. While NODE-SR [26] has been introduced to address the arbitrary scale super-resolution problem, our methodology marks the first application in image exposure enhancement. In contrast to NODE-SR [26], which learns the continuous variation of the scaling factor for the arbitrary scale super-resolution problem, our CLODE learns the continuous variation of image exposure through curve-adjustment.

3 Proposed Method

3.1 Preliminary

In photo editing applications, the curve-adjustment method is often used to adjust the tone of input images and provides effective exposure control. While this method is useful for pixel-wise manipulation, it is not well suited for images that contain areas of extreme over- or under-exposure. Additionally, a notable drawback of this approach is its reliance on manual adjustments (*e.g.*, the number of updates) by the user for each input image. This can be time-consuming and potentially less accurate in certain scenarios. To address this problem, Yuan and Sun [7] have proposed a solution that aims to mitigate the limitations of manual adjustments. They introduced an automated approach that involves estimating an image-specific *S*-shaped nonlinear tone curve (referred to as an *S*-curve) tailored to each input image. Specifically, for a given low-light image I_0, where each pixel value is in the range [0, 1], the *S*-curve formula for the enhanced image $I_0^{'}$ can be represented as follows:

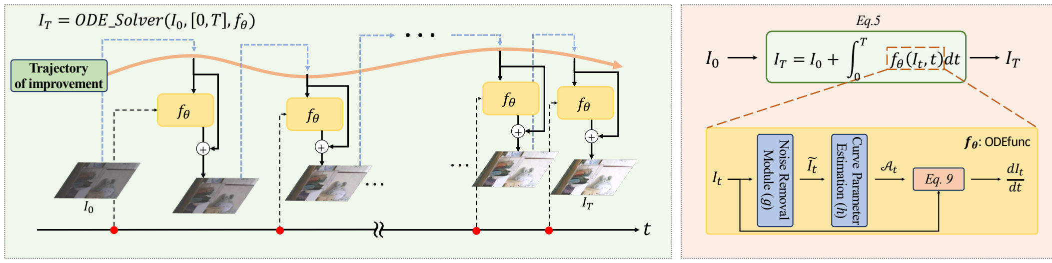
$$I_0^{'} = I_0 + \phi_s \cdot P_\Delta(I_0) - \phi_h \cdot P_\Delta(1 - I_0), \tag{1}$$

where ϕ_s and ϕ_h represent parameters for the amount of shadow and highlight, respectively. The function P_Δ serves as an increasing function for the adjustment that manipulates the intensity of individual pixels within the input of the function.

While Eq.1 allows for adjusting the brightness of an entire image using a single global curve parameter, existing iterative curve-adjustments approaches [6, 8, 9, 27] operate on a pixel-wise basis of the input images. Furthermore, they introduce the necessity of higher-order curves, which enhances images by fitting higher-order curves for fixed iteration steps while using a deep learning model to predict curve parameters on a pixel-by-pixel basis. Specifically the update formula enhances an image I_n at the n-th step to an image I_{n+1} at the next step as follows:

$$I_{n+1} = I_n + \mathcal{A}_n \otimes I_n \otimes (1 - I_n), \tag{2}$$

where $\mathcal{A}_n \in \mathbb{R}^{C \times H \times W}$ represents a pixel-wise varying curve parameter map and C, H, and W represent the number of channels, height, and width of the image I_n, and $\otimes$ operation denotes element-wise multiplication. Note that, the elements of $\mathcal{A}_n$ corresponding to the curve parameters at each



(a) Continuous update procedure of CLODE

(b) ODEfunc

Figure 2: **(a)** Illustration of continuous update procedure of CLODE. Optimal iterative update can be achieved through the ODE equation. **(b)** Illustration of our ODEfunc f_θ. ODEfunc contains the Noise Removal (g), Curve Parameter Estimation (h) module, and Eq. 9 to obtain the derivative value. Please refer to Appendix A.1.2 for more details.

pixel location are in the range $[-1, 1]$ and determine the quadratic curve for the pixel-wise exposure adjustment during the enhancement process. Conventional curve-adjustment methods [6, 8, 9, 18] iteratively follow this process for N times, fitting an appropriate higher-order curve to produce the final well-exposed output image. On the contrary, our CLODE performs curve adjustment for image enhancement by reformulating Eq. 2 as an ordinary differential equation. This approach facilitates memory-efficient training and yields more accurate results through adaptive computation using modern ODE solvers.

3.2 Continuous Exposure Learning for Low-light Image Enhancement using Neural ODEs

Although conventional curve-adjustment-based iterative methods offer advantages in terms of lightweight network architecture and local robustness, these approaches cannot guarantee convergence of the update process. ZeroDCE [6] empirically determines the iteration number N and enhances low-light images by iterating the curve-adjustment formula 8 (=N) times. While ReLLIE [9] provides users with optional flexibility, it requires manual selection of the value of N for each input image to further improve image quality. To tackle this challenge in optimization, we reformulate the curve-adjustment-based formula outlined in Eq. 2 as a Neural Ordinary Differential Equations (NODE). Then, we can solve the NODE with conventional ODE solvers (*e.g.*, Euler, RK4, dopri5) which guarantees the convergence of loss within tolerances. Specifically, we reformulate the original curve-adjustment-based formula by introducing a continuous state t instead of using the discrete state n as follows:

$$I_{t+1} = I_t + f_\theta(I_t, t), \tag{3}$$

where f_θ is a neural network with trainable parameters θ that satisfies $f_\theta(I_t, t) = \mathcal{A}_t \otimes I_t \otimes (1 - I_t)$. Then, we can parameterize the derivative of the enhanced image during the update using the network f_θ if the continuous update step is very small, and it is given by,

$$\frac{dI_t}{dt} = f_\theta(I_t, t). \tag{4}$$

By transforming the original curve fitting problem into a NODE problem with an initial condition I_0, we can estimate not only the derivative value of each state but also recover the enhanced image by solving the problem, and the initial value problem is given by,

$$I_T = I_0 + \int_0^T f_\theta(I_t, t)dt, \tag{5}$$

where I_T denotes the well-exposed image at the final state T. Finally, the low-light image enhancement process to output I_T is accomplished by using the ODE solver as:

$$I_T = \textbf{ODE\_Solver}(I_0, [0, T], f_\theta), \tag{6}$$

where **ODE_Solver** denotes a conventional algorithm for solving the ordinary differential equations. In our experiments, CLODE adopts the well-known dopri5 (Dormand-Prince 5th order Runge-Kutta) as an adaptive ODE solver, that determines an input-specific number of iterations for each input and dynamically adjusts the step size. Using the adaptive solver, we can adaptively compute the optimal state for different exposure levels, thereby enabling a more accurate approximation of the solution. This is in contrast to conventional methods, which use the same fixed number of iterations for all input images and cannot guarantee optimality and convergence. To the best of our knowledge, our approach is the first to define the low-light image enhancement problem as a novel NODE problem with an initial condition.

3.2.1 ODE function (ODEfunc)

We can solve the NODE problem in Eq. 5 by integrating f_θ over the time interval $[0, T]$ with the given initial value I_0 (e.g., a low-light image). In practice, conventional ODE solvers are used to address this problem, iteratively enhancing the low-light images using Eq. 3. In Fig. 2(a), we illustrate the continuous update procedure of our CLODE approach. Notably, the ODE function (ODEfunc) f_θ computes continuous dynamics of the latent image and is a key element in the update procedure. The detailed configuration of our ODEfunc f_θ is shown in Fig. 2(b). To be specific, our ODEfunc includes Noise Removal (g) and the Curve Parameter Estimation (h) modules with trainable parameters, and outputs $\frac{dI_t}{dt}$, the continuous dynamics of I_t. Please refer to Appendix A.1.2 for more details.

Noise Removal In the ODEfunc, we first employ a pre-processing step to eliminate the artifacts from I_t and generate the denoised image $\tilde{I}_t$ in order to produce more accurate curve adjustment parameters $\mathcal{A}_t$. To minimize computational costs within the f_θ, we employ a simple and lightweight three-layer convolutional neural network g as our Noise Removal module, expressed as follows:

$$\tilde{I}_t = g(I_t). \tag{7}$$

The refined image $\tilde{I}_t$ is then used as the input to the subsequent Curve Parameter Estimation stage.

Curve Parameter Estimation Inspired by [7, 28], to enhance both under- and over- exposed areas, we not only use the denoised image $\tilde{I}_t$ and its inverted version $(1 - \tilde{I}_t)$ as inputs to the Curve Parameter Estimation module. The formulation is given by:

$$\mathcal{A}_t = h(\tilde{I}_t, 1 - \tilde{I}_t), \tag{8}$$

where $\mathcal{A}_t$ represents the curve parameter map at t, and h represents the Curve Parameter Estimation module. For efficacy, this module is also a lightweight convolutional neural network. In particular, we apply layer normalization [29] to all intermediate features. Notably, the use of layer normalization enables CLODE to handle the diverse exposure ranges of input images. Furthermore, all convolutional layers within the Curve Parameter Estimation module h take the continuous state t as a conditional input, allowing for time-varying outputs during the integration interval $[0, T]$ as in [19].

Continuous Dynamics Lastly, the derivative value of the one-step state at t is computed in our ODEfunc, and it is expressed as follows:

$$\frac{dI_t}{dt} = \mathcal{A}_t \otimes I_t \otimes (1 - I_t). \tag{9}$$

Notably, unlike conventional curve-adjustment-based update formulas that discretize update steps, our continuous dynamics allows the desired level of accuracy and produces more accurate solutions.

3.3 Inference Process of CLODE

Inference Process Given a low-light input image I_0, CLODE undergoes successive image enhancement through f_θ until convergence within the specified tolerance of the ODE solvers, resulting in a well-exposed image I_T. Note that, the output image I_T may contain some noise that is amplified during the image enhancement process. Therefore, we use the noise-free image $\tilde{I}_T$ as our final outcome by applying the Noise Removal module g.

User Controllable Design CLODE learns the low-light exposure adjustment mechanism in the continuous-space, and is trained to output I_T by integrating the states from 0 to T in Eq. 5 using a fixed T. However, as shown in Fig. 3, users can manually adjust the integration interval by changing the final state value T at the test stage, allowing them to output images with the preferred exposure level and even produce images darker than the input. In practice, by controlling the final state from $-(T + \Delta t)$ to $(T + \Delta t)$, the exposure level of the output image can be easily controlled to provide a more user-friendly exposure level.

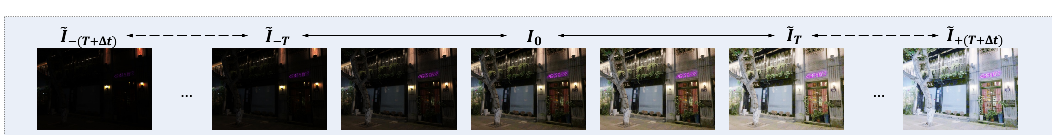


Figure 3: **Illustration of User Controllable Design**. By manually changing the integration interval from $-(T + \Delta t)$ to $+(T + \Delta t)$, ours can produce results with different exposure levels.

3.4 Zero-Reference Loss Functions

To address the challenge posed by the lack of ground truth, we use five zero-reference loss functions for unsupervised training.

Spatial Consistency Loss While the given low-light input image I_0 is enhanced during the update procedure, maintaining spatial consistency in the pixel brightness order is crucial for preserving image details. Specifically, we measure the difference in spatial consistency between the input image I_0 and our prediction I_T by comparing the differences in neighboring pixel values. Similar to [6], we compute the spatial consistency after applying 4-by-4 average pooling to both I_0 and I_T, and the spatial consistency loss $\mathcal{L}_{spa}$ is expressed as:

$$\mathcal{L}_{spa} = \frac{1}{K}\sum_{i=1}^{K}\sum_{j\in\Omega(i)}(|m_4(I_T)_i - m_4(I_T)_j| - |m_4(I_0)_i - m_4(I_0)_j|)^2. \tag{10}$$

The 4-by-4 average pooling operation is denoted as $m_4(\cdot)$ and $\Omega(i)$ includes neighboring pixels in four directions (left, right, top, bottom) centered at position i. The normalization factor K denotes the number of pixels in the reduced image after the pooling operation, and K is given by $\frac{H}{4} \times \frac{W}{4} \times C$.

Exposure Loss To enforce a consistent exposure level across pixels, conventional unsupervised methods incorporate exposure guidance into the loss function [6]. Similarly, we introduce a desired exposure level parameter E and define the exposure loss $\mathcal{L}_{exp}$ as:

$$\mathcal{L}_{exp} = ||m_{16}(I_T) - \mathrm{E}||_2^2. \tag{11}$$

In our experiments, we set the exposure level E to 0.6, which corresponds to the gray level in the RGB color space. To maintain the overall exposure level in the results, we minimize the difference between the pixel values of the predicted image I_T and the desired exposure level E after performing a 16-by-16 average pooling operation $m_{16}(\cdot)$ on the output image I_T.

Color Constancy Loss In conventional zero-reference methods, two main approaches are used to enforce spatial color constancy: one based on the retinex-theory, and the other based on the Gray-World hypothesis in [30]. In this work, the color constancy loss $\mathcal{L}_{col}$ is based on the Gray-World hypothesis as in [6, 15], and the formulation is given by,

$$\mathcal{L}_{col} = (R - B)^2 + (R - G)^2 + (G - B)^2, \tag{12}$$

where R, G, and B are the mean pixel values of the red, green, and blue channels in the predicted image I_T, respectively. We minimize the color constancy loss $\mathcal{L}_{col}$ to correct the potential color deviations in the enhanced image.

Parameter Regularization Loss To prevent rapid changes of pixel values in nearby regions, we employ the spatial regularization to enforce smoothness among neighboring curve parameter values in $\mathcal{A}_t$, and the formulation is given by,

$$\mathcal{L}_{param} = (|\nabla_x\mathcal{A}_0| + |\nabla_y\mathcal{A}_0|)^2 + \ldots + (|\nabla_x\mathcal{A}_{T-1}| + |\nabla_y\mathcal{A}_{T-1}|)^2, \tag{13}$$

where the linear operations ∇_x and ∇_y compute the horizontal and vertical gradients from the parameter map $\mathcal{A}_t$, respectively. For better understanding, we represent $T - 1$ as the stage before the final enhancement. We employ the parameter regularization loss at each update step (*e.g.*, red points in Fig. 2 (a)) and accumulate the loss while solving the NODE problem.

Noise Removal Loss To estimate a spatially smooth $\mathcal{A}_t$ regardless of the noise in the image I_t, we use the Noise Removal module (g) to remove the noise. To train the Noise Removal module, we utilize a self-supervision-based loss $\mathcal{L}_{noise}$ that follows the Noise2Noise approaches [31, 32, 33]. Specifically, we employ the loss introduced in Zeroshot-N2N [33]. Our $\mathcal{L}_{noise}$ has two components at state t: the residual loss $\mathcal{L}^t_{res}$ and the consistency loss $\mathcal{L}^t_{cons}$. We minimize these losses using two different down-samplers; D_1 and D_2. Notably, D_1 and D_2 represent fixed 2D convolutional kernels: $\begin{bmatrix}0.5 & 0\\ 0 & 0.5\end{bmatrix}$ and $\begin{bmatrix}0 & 0.5\\ 0.5 & 0\end{bmatrix}$, respectively. Notably, these kernels are used for downsampling through convolutions with a stride of two. First, our $\mathcal{L}^t_{res}$ fits the noise within I_t through a symmetric loss function similar to the approach in [34] and it yields:

$$\mathcal{L}^t_{res} = \frac{1}{2}(||D_1(I_t) - g(D_1(I_t)) - D_2(I_t)||_2^2 + ||D_2(I_t) - g(D_2(I_t)) - D_1(I_t)||_2^2). \tag{14}$$

Next, as in [33], $\mathcal{L}^t_{cons}$ ensures spatial consistency by maintaining similarity in noise distributions, even if the order of denoising and downsampling is altered. Specifically, $\mathcal{L}^t_{cons}$ also adopts a symmetric loss and is defined as at each update step (*e.g.*, red points in Fig. 2 (a)):

$$\mathcal{L}^t_{cons} = \frac{1}{2}(||D_1(I_t) - g(D_1(I_t)) - D_1(I_t - g(I_t))||_2^2 + ||D_2(I_t) - g(D_2(I_t)) - D_2(I_t - g(I_t))||_2^2). \tag{15}$$

Therefore, our final noise removal loss $\mathcal{L}_{noise}$ can be represented accumulating during the update procedure as:

$$\mathcal{L}_{noise} = (\mathcal{L}^0_{res} + \mathcal{L}^0_{cons}) + \ldots + (\mathcal{L}^{T-1}_{res} + \mathcal{L}^{T-1}_{cons}). \tag{16}$$

As with Eq. 13, we represent $T-1$ as the stage before the final enhancement. A more detailed description of the noise removal loss is provided in Appendix A.4.

Final Objective Function The final objective function to optimize is given as follows:

$$\mathcal{L}_{total} = w_{spa} \cdot \mathcal{L}_{spa} + w_{exp} \cdot \mathcal{L}_{exp} + w_{col} \cdot \mathcal{L}_{col} + w_{param} \cdot \mathcal{L}_{param} + w_{noise} \cdot \mathcal{L}_{noise}, \tag{17}$$

where w_{spa}, w_{exp}, w_{col}, w_{param}, and w_{noise} are hyper-parameters used to control the relative significance of each associated loss during the training process.

4 Experiments

4.1 Implementation Details

Please refer to Appendix A.1 for more implementation details and training scheme. The code will be available upon acceptance.

4.2 Experimental Setup

In this work, we use the LOL [5] and SICE [35] Part1 datasets for training. The results of low-light image enhancement are evaluated on the LOL and LSRW [4] benchmark datasets. In addition, the SICE [35] Part2 dataset is used as a benchmark dataset for evaluation under various exposure conditions. SICE Part2 contains 229 image sequences with different exposure levels, and we use the entire sequences as the evaluation dataset. By default, each comparison model uses its official network weights. In cases where the official code is available but weights are not provided, the models are retrained using the official code and settings, except for ReLLIE [9]. We present the performance of ReLLIE on the LOL dataset as reported in their original manuscript.

4.3 Quantitative Comparisons

First, we quantitatively compare the performance of low-light image enhancement on different datasets. Notably, in the experimental results, CLODE represents our proposed method without requiring additional user input (by default), while CLODE† represents the result of adjusting the final state T to the user's preferred level, as introduced in Sec. 3.3.

In Table 1, we compare the low-light image enhancement performance on the LSRW [4] and LOL [5] benchmark datasets in terms of peak signal-to-noise ratio (PSNR) and structural similarity (SSIM). The term "*GT Mean*" refers to the evaluation method used by KinD [36] and LLFlow [1], which matches the average value of the output pixels to that of the ground truth pixels. CLODE and CLODE† outperform other unsupervised learning methods. Notably, CLODE† even surpasses the PSNR of state-of-the-art supervised learning methods by 0.73 dB, when averaging the results from the LSRW and LOL datasets in the rightmost columns, without using *GT Mean*. Moreover, two notable points can be highlighted in Table 1. First, the effectiveness of using NODE to compute accurate higher order curves is evident, as demonstrated by its superiority over curve-adjustment-based methods; ZeroDCE [6] and ReLLIE [9]. Second, unlike other models trained on the same training dataset (LOL), our model shows robust performance on both the LSRW and LOL test datasets, indicating that our model generalizes better than conventional approaches.

In Table 2, we demonstrate the robustness under various exposure conditions including both under- and over- exposures, and evaluate the performance on SICE Part2 [35]. The results show that CLODE exhibits robust performance compared to other models, even under various exposure conditions. It outperforms other unsupervised learning methods, and even when compared to supervised learning

Table 1: **Quantitative results on LSRW [4] and LOL [5] datasets**. For a fair comparison, we re-trained some models on LOL and marked them with *. Among the unsupervised approaches, the best score is displayed in **red**, the second best in **blue**, and the third best in **black**. For more comparison results in terms of non-reference metrics, please refer to Appendix A.4.3.

Training	Method	#Params (M)	Train dataset	LSRW Normal PSNR↑	LSRW Normal SSIM↑	LSRW *GT Mean* PSNR↑	LSRW *GT Mean* SSIM↑	LOL Normal PSNR↑	LOL Normal SSIM↑	LOL *GT Mean* PSNR↑	LOL *GT Mean* SSIM↑	Average Normal PSNR↑	Average Normal SSIM↑	Average *GT Mean* PSNR↑	Average *GT Mean* SSIM↑
Supervised	RetinexNet [5]	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 [37]	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 [38]	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 [36]	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 [1]	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 [2]	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 [11]	0.0003	MIT-5K [39]	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 [11]	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 [11]	0.0003	DARKFace [40]	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* [11]	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 [10]	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* [6]	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 [9]	-	LOL	-	-	-	-	18.37	0.641	-	-	-	-	-	-
	PairLIE [13]	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 [17]	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**

Table 2: **Quantitative results on SICE [35] 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↑	Normal SSIM↑	Normal LPIPS↓	Normal NIQE↓	Normal BRISQUE↓	Normal PI↓	Normal Entropy↑	*GT Mean* PSNR↑	*GT Mean* SSIM↑
Supervised	URetinexNet [37]	LOL, SICE	12.15	0.708	0.393	4.250	15.633	3.372	6.926	17.81	0.686
	LLFlow* [1]	SICE	14.34	0.608	0.279	3.643	17.011	3.481	6.566	19.59	0.658
	ECLNet [41]	SICE	13.99	0.562	0.290	4.279	24.570	3.520	6.919	16.66	0.690
	FECNet [42]	SICE	14.25	0.600	0.291	3.786	17.454	3.025	7.035	16.47	0.639
	RetinexFormer* [2]	SICE	19.12	0.570	0.369	4.452	24.768	4.573	7.025	20.97	0.578
	RetinexFormer [2]	MIT-5K [39]	13.23	0.564	0.263	3.848	17.350	2.863	6.881	16.35	0.609
Unsupervised	SCI-easy [11]	MIT-5K [39]	9.87	0.486	0.372	4.276	21.850	3.226	6.113	16.44	0.622
	SCI-medium [11]	LOL, LSRW	9.77	0.510	0.454	5.727	33.200	4.392	5.212	15.83	0.574
	SCI-difficult [11]	DarkFace [40]	11.13	0.577	0.324	4.636	23.620	3.107	6.386	16.85	0.647
	SCI* [11]	SICE	10.67	0.478	0.331	4.289	23.449	3.570	6.213	17.99	0.675
	RUAS* [10]	SICE	9.12	0.408	0.539	8.097	52.923	6.004	5.101	15.52	0.531
	ZeroDCE [6]	SICE	12.67	**0.635**	**0.244**	**3.886**	**21.630**	**2.821**	6.516	18.85	**0.686**
	PairLIE [13]	LOL, SICE	**13.39**	0.619	0.305	5.268	36.536	3.548	6.376	19.22	0.663
	Night-Enhancement* [17]	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**

methods, CLODE† and CLODE achieve the best and second best results, respectively. Despite being an unsupervised method, CLODE narrows the performance gap with state-of-the-art supervised methods. Additionally, it operates robustly under challenging conditions such as various exposure conditions in SICE Part2, surpassing supervised approaches. These strengths distinguish CLODE from other unsupervised learning methods.

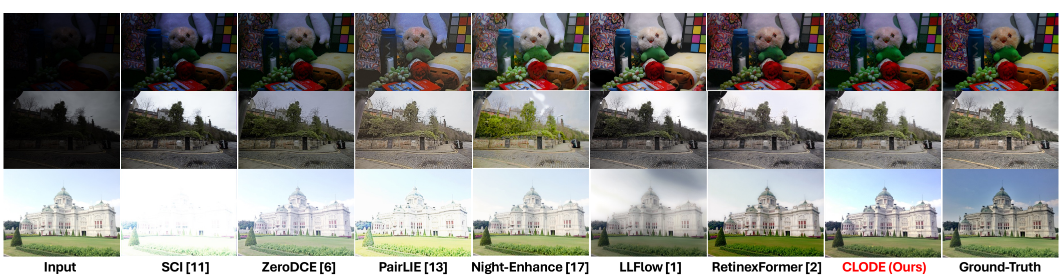


Figure 4: **Visual comparisons.** From top to bottom: LOL [5], under- and over-exposed image of the SICE [35] Part2. For more visual results, please refer to Fig. 10 in the Appendix.

4.4 Perceptual and Visual Comparisons

In Table 2, we also provide a perceptual comparison of the results with other methods. The evaluation is conducted on SICE Part2, which includes a combination of underexposed and overexposed images. To measure the perceptual quality, we adopt Learned Perceptual Image Patch Similarity (LPIPS) [43], and non-reference metrics; natural image quality evaluator (NIQE) [44], blind/referenceless image spatial quality evaluator (BRISQUE) [45], perception index (PI) [46], and Entropy [47]. In these four aspects, both CLODE and CLODE† show outstanding performance compared to existing unsupervised methods. The visual results are compared in Fig. 4. CLODE shows robust and natural image enhancement results compared to other comparison methods, regardless of the exposure conditions of the input image.

Table 3: Comparative experiments according to using NODE on LSRW [4]/LOL [5].The "Discrete" refers to performing curve adjustment in discrete steps, similar to the conventional methods [6, 9], 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**

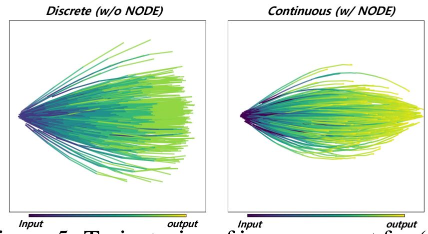


Figure 5: Trajectories of improvement for (e1) and (f1) in Table 3. PCA dimension reduction is used to visualize the trajectories.

4.5 Ablation Study

Effectiveness of NODE To validate the impact of NODE, we adjust the curves using the architecture of CLODE in both discrete (w/o NODE) and continuous (w/ NODE) spaces, and we compare the results in Table 3. In the discrete setting, similar to [6], curve parameters for fixed update steps [1, 5, 10, 20, 30] are estimated in parallel ((a1) - (e1)). In the continuous setting, however, curve parameters are estimated sequentially for non-fixed adaptive steps, up to a maximum of 30 steps ((f1)). Results in Table 3 demonstrate that the curve parameters produced during the sequential continuous update procedure are more accurate and verify superior performance of the proposed method over the update procedure in the conventional discrete setting. In addition, in Fig. 5, we visualize the trajectories of improvement by plotting PCA dimension reduction results of latent images during updates. We observe that when curve adjustments are made in continuous space by (f1), the trajectories converge more accurately at the final states compared to (e1). This demonstrates that using NODE to find the optimal state certainly contributes to image enhancement.

Table 4: Impact of the modules in f_θ. Noise Removal and the layer normalization (LN) significantly improve performance.

Case	Noise Removal g	LN in h	PSNR↑	SSIM↑
(a2)			14.72	0.538
(b2)	✓		15.19	0.489
(c2)		✓	18.67	0.577
(d2)	✓	✓	19.61	0.718

Table 5: Execution time and performance.

Training	Method	PSNR/SSIM	#Params (M)	Time (S)
Supervised	RetinexNet [5]	15.49/0.355	0.4446	0.337
	LLFlow [1]	17.52/0.509	38.859	0.144
	RetinexFormer [2]	17.76/0.517	1.6057	0.072
Unsupervised	SCI-medium [11]	15.24/0.424	0.0003	0.001
	RUAS [10]	14.27/0.470	0.0034	0.006
	ZeroDCE [6]	15.81/0.449	0.0794	0.004
	PairLIE [13]	16.97/0.498	0.3417	0.008
	CLODE	17.28/0.533	0.2167	0.056
	CLODE-S	16.97/0.457	0.0004	0.005

Effect of the Modules In Table 4, we conduct ablation experiments on the modules used in ODEfunc f_θ. We verify the effects of the Noise Removal module g and the layer normalization (LN) in the Curve Parameter Estimation module h. Each module shows performance improvements compared to the baseline (a2). In particular, our final model (d2) achieves the largest performance gain in terms of PSNR/SSIM. Furthermore, case (c2), which includes layer normalization, has about a 4dB gain in PSNR compared to (a2), which does not include layer normalization. This shows that during the image enhancement process in NODE, it is essential to use layer normalization to normalize each state. The visual results can be seen in Fig. 8 of the Appendix.

5 Limitations

Table 5 shows the performance of PSNR/SSIM, number of parameters, and execution time measured in LSRW [4] using an NVIDIA RTX 4090. CLODE shows the advantage in model size compared to supervised methods. The iterative ODE solving process of CLODE takes longer than lightweight unsupervised models, but it exhibits faster speed and performance comparable to supervised methods. Additionally, a smaller version, CLODE-S in Appendix A.1.2 shows promising enhancement performance with competitive inference time comparable to those of unsupervised models.

6 Conclusions

In this work, we address the unsupervised low-light image enhancement problem by formulating it as a NODE problem. We introduce a novel iterative curve-adjustment approach with NODE, transforming discrete iterative problems into continuous ones. CLODE exhibits superior convergence compared to other unsupervised iterative methods, especially in diverse low-light and multi-exposure scenarios. In conclusion, our method effectively narrows the performance gap between unsupervised and supervised methods across various datasets.

References

[1] Yufei Wang, Renjie Wan, Wenhan Yang, Haoliang Li, Lap-Pui Chau, and Alex Kot. Low-light image enhancement with normalizing flow. In *AAAI*, volume 36, 2022.

[2] Yuanhao Cai, Hao Bian, Jing Lin, Haoqian Wang, Radu Timofte, and Yulun Zhang. Retinexformer: One-stage retinex-based transformer for low-light image enhancement. 2023.

[3] Jinhui Hou, Zhiyu Zhu, Junhui Hou, Hui Liu, Huanqiang Zeng, and Hui Yuan. Global structure-aware diffusion process for low-light image enhancement. *NeurIPS*, 2023.

[4] Jiang Hai, Zhu Xuan, Ren Yang, Yutong Hao, Fengzhu Zou, Fang Lin, and Songchen Han. R2rnet: Low-light image enhancement via real-low to real-normal network. *Journal of Visual Communication and Image Representation*, 90, 2023.

[5] Wenhan Yang Jiaying Liu Chen Wei, Wenjing Wang. Deep retinex decomposition for low-light enhancement. In *BMVC*, 2018.

[6] Chunle Guo, Chongyi Li, Jichang Guo, Chen Change Loy, Junhui Hou, Sam Kwong, and Runmin Cong. Zero-reference deep curve estimation for low-light image enhancement. In *CVPR*, 2020.

[7] Lu Yuan and Jian Sun. Automatic exposure correction of consumer photographs. In *ECCV*. Springer, 2012.

[8] Chongyi Li, Chunle Guo, and Chen Change Loy. Learning to enhance low-light image via zero-reference deep curve estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44, 2021.

[9] Rongkai Zhang, Lanqing Guo, Siyu Huang, and Bihan Wen. Rellie: Deep reinforcement learning for customized low-light image enhancement. In *ACMMM*, 2021.

[10] Risheng Liu, Long Ma, Jiaao Zhang, Xin Fan, and Zhongxuan Luo. Retinex-inspired unrolling with cooperative prior architecture search for low-light image enhancement. In *CVPR*, 2021.

[11] Long Ma, Tengyu Ma, Risheng Liu, Xin Fan, and Zhongxuan Luo. Toward fast, flexible, and robust low-light image enhancement. In *CVPR*, 2022.

[12] Zunjin Zhao, Bangshu Xiong, Lei Wang, Qiaofeng Ou, Lei Yu, and Fa Kuang. Retinexdip: A unified deep framework for low-light image enhancement. *IEEE Transactions on Circuits and Systems for Video Technology*, 32, 2021.

[13] Zhenqi Fu, Yan Yang, Xiaotong Tu, Yue Huang, Xinghao Ding, and Kai-Kuang Ma. Learning a simple low-light image enhancer from paired low-light instances. In *CVPR*, 2023.

[14] Qiuping Jiang, Yudong Mao, Runmin Cong, Wenqi Ren, Chao Huang, and Feng Shao. Unsupervised decomposition and correction network for low-light image enhancement. *IEEE Transactions on Intelligent Transportation Systems*, 23(10), 2022.

[15] Feng Zhang, Yuanjie Shao, Yishi Sun, Kai Zhu, Changxin Gao, and Nong Sang. Unsupervised low-light image enhancement via histogram equalization prior. *arXiv preprint arXiv:2112.01766*, 2021.

[16] Yifan Jiang, Xinyu Gong, Ding Liu, Yu Cheng, Chen Fang, Xiaohui Shen, Jianchao Yang, Pan Zhou, and Zhangyang Wang. Enlightengan: Deep light enhancement without paired supervision. *IEEE Transactions on image processing*, 30, 2021.

[17] Yeying Jin, Wenhan Yang, and Robby T Tan. Unsupervised night image enhancement: When layer decomposition meets light-effects suppression. In *ECCV*, 2022.

[18] Chongyi Li, Chunle Guo, Ruicheng Feng, Shangchen Zhou, and Chen Change Loy. Cudi: Curve distillation for efficient and controllable exposure adjustment. *arXiv preprint arXiv:2207.14273*, 2022.

[19] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *NeurIPS*, 31, 2018.

[20] Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural odes. *NeuraIPS*, 32, 2019.

[21] Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *NeuraIPS*, 32, 2019.

[22] Yogesh Verma, Markus Heinonen, and Vikas Garg. Climode: Climate forecasting with physics-informed neural odes. In *ICLR*, 2023.

[23] Sunghyun Park, Kangyeol Kim, Junsoo Lee, Jaegul Choo, Joonseok Lee, Sookyung Kim, and Edward Choi. Vid-ode: Continuous-time video generation with neural ordinary differential equation. In *AAAI*, volume 35, 2021.

[24] Yifan Wu, Tom Z Jiahao, Jiancong Wang, Paul A Yushkevich, M Ani Hsieh, and James C Gee. Nodeo: A neural ordinary differential equation based optimization framework for deformable image registration. In *CVPR*, 2022.

[25] Boyan Jiang, Yinda Zhang, Xingkui Wei, Xiangyang Xue, and Yanwei Fu. Learning compositional representation for 4d captures with neural ode. In *CVPR*, 2021.

[26] Seobin Park and Tae Hyun Kim. Progressive image super-resolution via neural differential equation. In *ICASSP*, 2022.

[27] Jiancheng Huang, Yifan Liu, and Shifeng Chen. Bootstrap diffusion model curve estimation for high resolution low-light image enhancement. In *PRICAI*. Springer, 2023.

[28] Haoyuan Wang, Ke Xu, and Rynson WH Lau. Local color distributions prior for image enhancement. In *ECCV*. Springer, 2022.

[29] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[30] Gershon Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin institute*, 310, 1980.

[31] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2Noise: Learning image restoration without clean data. In *ICML*, 2018.

[32] Tao Huang, Songjiang Li, Xu Jia, Huchuan Lu, and Jianzhuang Liu. Neighbor2neighbor: Self-supervised denoising from single noisy images. In *CVPR*, 2021.

[33] Youssef Mansour and Reinhard Heckel. Zero-shot noise2noise: Efficient image denoising without any data. In *CVPR*, 2023.

[34] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021.

[35] Jianrui Cai, Shuhang Gu, and Lei Zhang. Learning a deep single image contrast enhancer from multi-exposure images. *IEEE Transactions on Image Processing*, 27, 2018.

[36] Yonghua Zhang, Jiawan Zhang, and Xiaojie Guo. Kindling the darkness: A practical low-light image enhancer. In *ACMMM*, 2019.

[37] Wenhui Wu, Jian Weng, Pingping Zhang, Xu Wang, Wenhan Yang, and Jianmin Jiang. Uretinex-net: Retinex-based deep unfolding network for low-light image enhancement. In *CVPR*, 2022.

[38] Wenhan Yang, Shiqi Wang, Yuming Fang, Yue Wang, and Jiaying Liu. Band representation-based semi-supervised low-light image enhancement: Bridging the gap between signal fidelity and perceptual quality. *IEEE Transactions on image processing*, 2021.

[39] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input / output image pairs. In *CVPR*, 2011.

[40] Wenhan Yang, Ye Yuan, Wenqi Ren, Jiaying Liu, Walter J Scheirer, Zhangyang Wang, Taiheng Zhang, Qiaoyong Zhong, Di Xie, Shiliang Pu, et al. Advancing image understanding in poor visibility environments: A collective benchmark study. *IEEE Transactions on image processing*, 29, 2020.

[41] Jie Huang, Man Zhou, Yajing Liu, Mingde Yao, Feng Zhao, and Zhiwei Xiong. Exposure-consistency representation learning for exposure correction. In *ACMMM*, 2022.

[42] Jie Huang, Yajing Liu, Feng Zhao, Keyu Yan, Jinghao Zhang, Yukun Huang, Man Zhou, and Zhiwei Xiong. Deep fourier-based exposure correction network with spatial-frequency interaction. In *ECCV*, 2022.

[43] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[44] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal processing letters*, 20, 2012.

[45] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on image processing*, 21, 2012.

[46] Yochai Blau, Roey Mechrez, Radu Timofte, Tomer Michaeli, and Lihi Zelnik-Manor. The 2018 pirm challenge on perceptual image super-resolution. In *ECCVW*, 2018.

[47] Xiaoqiao Chen, Qingyi Zhang, Manhui Lin, Guangyi Yang, and Chu He. No-reference color image quality assessment: From entropy to perceptual quality. *EURASIP Journal on Image and Video Processing*, 2019.

[48] Ricky T. Q. Chen. torchdiffeq, 2018.

[49] Mahmoud Afifi, Konstantinos G Derpanis, Bjorn Ommer, and Michael S Brown. Learning multi-scale photo exposure correction. In *CVPR*, 2021.

A Appendix

A.1 Implement Details

The training set of images is resized to 128x128, we employ the *Pytorch* framework on NVIDIA A6000 GPU with a batch size of 8. The ADAM optimizer is used with default parameters and a fixed learning rate of $1e^{-5}$ to optimize the parameters of our network. The weights for the loss function $w_{col}, w_{param}, w_{spa}, w_{exp}$ and w_{noise} are set to 20, 200, 1, 10 and 1 respectively, to balance the scale of losses. Furthermore, we adopt *torchdiffeq* [48] for Neural ODEs implementation. The training process is conducted for 100 epochs.

A.1.1 Implementation details of NODE

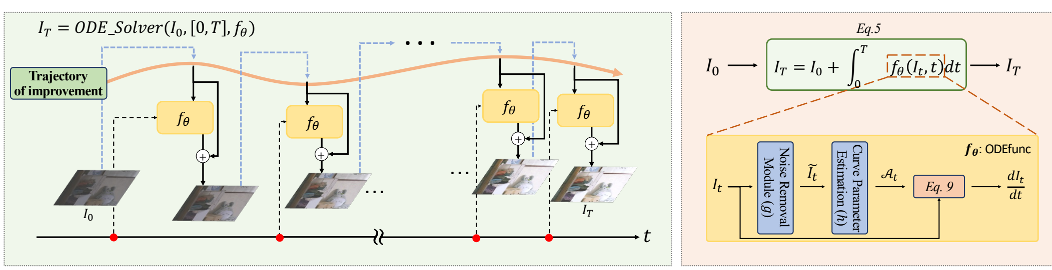
We utilize the adaptive ODE solver, dopri5 (Dormand-Prince Runge-Kutta of Order 5) for our work. The maximum allowed step for the adaptive solver is set to 30. The relative and absolute tolerances for the error rate calculation are set uniformly to $1e^{-5}$. The adaptive solver uses the error rate to determine the steps. Also, the adaptive solver estimates an error rate for the current step t, and if the error exceeds the allowable tolerances, the step is re-done with a smaller step size. This process continues until the error becomes smaller than the provided tolerance. The error rate Γ_t at the t-th step is computed as follows:

$$\Gamma_t = atol \times rtol \times norm(I_t), \tag{18}$$

where $atol$ is absolute tolerance, and $rtol$ is relative tolerate, and the norm being used is a mixed L-infinity/RMS norm. We set both $atol$ and $rtol$ to 1e-5.

A.1.2 Details of the CLODE architecture

This section presents the architectural details of the CLODE network architecture, with a particular focus on the ODEfunc module. The Noise Removal module g employs a simple and lightweight three-layer convolutional network. In Curve Parameter Estimation module h, a shallow network with two branches is utilized, wherein filters of varying sizes are employed at each branch to capture image features across different filter scales. We also provide architectural details of CLODE-S as mentioned in Sec. 5 of the main manuscript. This version omits the Noise Removal module for speed and uses a 2-layer network with 1x1 convolutions.



(a) Continuous update procedure of CLODE

(b) ODEfunc

Figure 6: Illustration of architecture details of (a) modules of ODEfunc in CLODE and (b) ODEfunc of CLODE-S.

A.2 Impact of Each Loss Functions

CLODE combines five non-reference loss functions to train NODE, producing optimal improvements. We present ablation experiments for each loss function, and the results are presented in Table 6 and Fig. 7. The results of each image ablation experiment demonstrate that appropriate improvement results can only be obtained when using CLODE with all loss functions. The characteristics of the loss function as observed in each ablation are as follows: **((a3) w/o $\mathcal{L}_{exp}$)**: Brightness improvement is not achieved in low-exposure enhancement. **((b3) w/o $\mathcal{L}_{col}$)**: Severe color distortion occurs in over-exposure enhancement, damaging structural details. **((c3) w/o $\mathcal{L}_{param}$)**: Structural distortion occurs, creating artifacts. **((d3) w/o $\mathcal{L}_{spa}$)**: While showing better results than other experiments, it

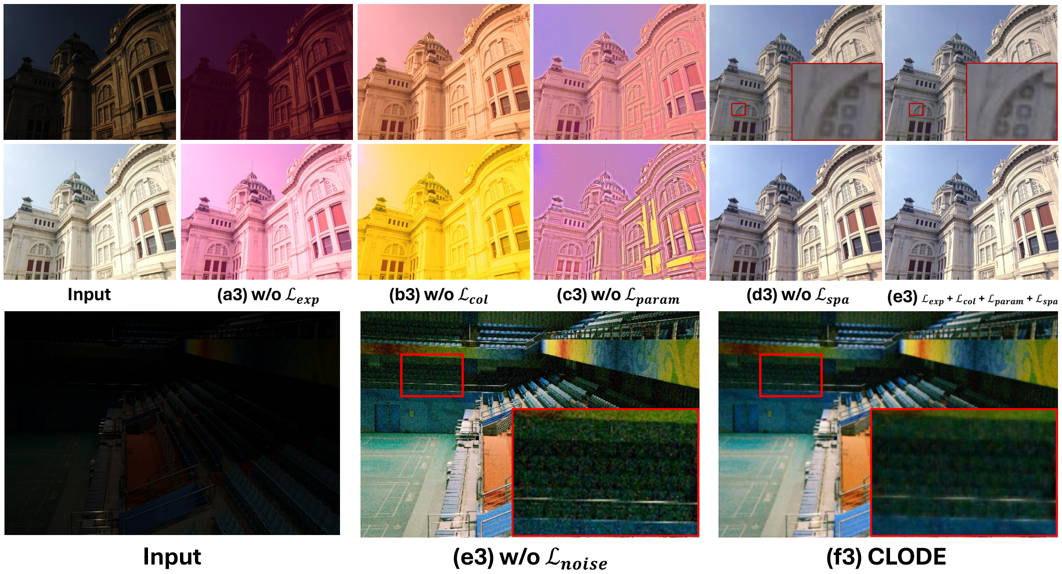


Figure 7: Visual results for the ablation study of each loss function. CLODE combines five non-reference loss functions in training for producing optimal enhancement results.

Table 6: Ablation study on each non-reference losses. The experiment is evaluated on LOL [5].

Case	$\mathcal{L}_{spa}$	$\mathcal{L}_{exp}$	$\mathcal{L}_{col}$	$\mathcal{L}_{param}$	$\mathcal{L}_{noise}$	PSNR	SSIM
(a3)	✓		✓	✓		8.84	0.323
(b3)	✓	✓		✓		14.72	0.566
(c3)	✓	✓	✓			14.76	0.535
(d3)		✓	✓	✓		18.76	0.580
(e3)	✓	✓	✓	✓		18.92	0.582
(f3)	✓	✓	✓	✓	✓	**19.61**	**0.718**

occurs loss of structural details compared to (e3). **((e3) w/o $\mathcal{L}_{Noise}$)**: Compared to the proposed version (f3), it produces improved results with noise present.

A.3 Visualization of curve parameter map $\mathcal{A}$

We provide visual comparison results for the module ablation experiments in Sec. 4.5 of the main manuscript. In the visual results without noise removal module (c2), we can observe the noise in $\mathcal{A}$. The enhanced result of (c2) using $\mathcal{A}$ with noise shows overall color discrepancy compared to the ground-truth, in contrast to the enhanced result of (d2) where the noise removal module are applied. The enhanced result of (d2) shows robust color similarity with the ground-truth image. We can confirm that removing noise for $\mathcal{A}$ is important for curve-adjustment-based method.

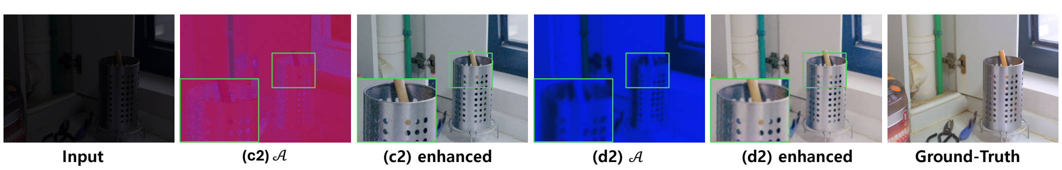


Figure 8: A visual comparison of the results for (c2) and (d2) from Table 4 in the main manuscript. The enhanced result (d2) using $\mathcal{A}$ with noise removal module demonstrates improvement more similar to the ground-truth.

A.4 Background of Noise Removal Loss

In Sec.3.4 we provide information about the zero-reference loss functions that we used. Unlike the others, the Noise Removal Loss ($\mathcal{L}_{noise}$) requires more explanation due to its unfamiliarity in the field of low-light enhancement, so we provide additional explanation for it.

A.4.1 Noise2Noise background

In supervised denoising studies, neural networks are aimed at denoising the noisy image $\mathbf{y}$ to the clean image $\mathbf{x}$. Since the noisy $\mathbf{y}$ is an addition of the clean image $\mathbf{x}$ and the noise $\mathbf{e}$, the network is trained to map the noise $\mathbf{e}$ which is called Noise2Clean (N2C) method. If the network parameter is ϕ_{N2C}, the object function of the supervised denoising method with the network g_ϕ can be written as:

$$\phi_{N2C} = \arg\min_{\phi} \mathbb{E}\big[||g_\phi(\mathbf{y}) - \mathbf{x}||_2^2\big]. \tag{19}$$

Denoising networks can also be trained to output the noisy image $\mathbf{y}_2$ from the noisy input image $\mathbf{y}_1$ that comes from the same clean image $\mathbf{x}$. This noise-to-noise manner can be achieved by assuming that the noise has a mean of zero as introduced in Noise2Noise (N2N) [31]. This is the objective function for the N2N network parameter ϕ_{N2N}:

$$\phi_{N2N} = \arg\min_{\phi} \mathbb{E}\big[||g_\phi(\mathbf{y}_2) - \mathbf{y}_1||_2^2\big]. \tag{20}$$

The N2N manner shows close performance compare to N2C manner with sufficient training data since the objective functions of N2C and N2N are aimed on the same network parameter. If $\mathbf{y}_a = \mathbf{x} + \mathbf{e}_a$, $\mathbf{y}_b = \mathbf{x} + \mathbf{e}_b$, and the mean value of $\mathbf{e}_a$ and $\mathbf{e}_b$ are zero, the proof is as follows:

$$\begin{aligned}
\phi_{N2C} &= \arg\min_{\phi} \mathbb{E}\big[||g_\phi(\mathbf{y}_2) - \mathbf{x}||_2^2\big] \\
&= \arg\min_{\phi} \mathbb{E}\big[||g_\phi(\mathbf{y}_2)||_2^2 - 2\mathbf{x}^\intercal g_\phi(\mathbf{y}_2) + ||\mathbf{x}||_2^2\big] \\
&= \arg\min_{\phi} \mathbb{E}\big[||g_\phi(\mathbf{y}_2)||_2^2 - 2\mathbf{x}^\intercal g_\phi(\mathbf{y}_2)\big] \\
\phi_{N2N} &= \arg\min_{\phi} \mathbb{E}\big[||g_\phi(\mathbf{y}_2) - \mathbf{y}_1||_2^2\big] \\
&= \arg\min_{\phi} \mathbb{E}\big[||g_\phi(\mathbf{y}_2) - (\mathbf{x} + \mathbf{e}_1)||_2^2\big] \\
&= \arg\min_{\phi} \mathbb{E}\big[||g_\phi(\mathbf{y}_2)||_2^2 - 2\mathbf{x}^\intercal g_\phi(\mathbf{y}_2) - 2\mathbf{e}_2^\intercal g_\phi(\mathbf{y}_2) + ||\mathbf{x} + \mathbf{e}_1||_2^2\big] \\
&= \arg\min_{\phi} \mathbb{E}\big[||g_\phi(\mathbf{y}_2)||_2^2 - 2\mathbf{x}^\intercal g_\phi(\mathbf{y}_2) - 2\mathbf{e}_2^\intercal g_\phi(\mathbf{y}_2)\big] \\
&= \arg\min_{\phi} \mathbb{E}\big[||g_\phi(\mathbf{y}_2)||_2^2 - 2\mathbf{x}^\intercal g_\phi(\mathbf{y}_2)\big].
\end{aligned} \tag{21}$$

By Eq. 21 we can confirm that the object of ϕ_{N2C} and ϕ_{N2N} is the identical one.

A.4.2 Zeroshot Noise2Noise method

In spite of N2N approaches, it is hard to obtain two different noisy images from the same clean scene. To address this hurdle, the Neighbor2Neighbor [32] method is proposed. This allows a pair of noisy images to be augmented from a single noisy image coming from the same clean image. In Zeroshot-N2N [33], which is adopted in our proposed method, Neighbor2Neighbor is achieved by using two different 2D convolutional kernels (D_1 and D_2) on noisy images. If the noisy image is $\mathbf{y}$, a pair of down-sampled images $\mathbf{y}_1$, $\mathbf{y}_2$ can be represented as:

$$\mathbf{y}_1 = D_1(\mathbf{y}), \mathbf{y}_2 = D_2(\mathbf{y}). \tag{22}$$

For a noisy image $\mathbf{y}$ with a size of $H \times W \times C$, the size of $\mathbf{y}_1$ and $\mathbf{y}_2$ is $\frac{H}{2} \times \frac{W}{2} \times C$. With downsampled images $\mathbf{y}_1$ and $\mathbf{y}_2$, the loss optimizes g_ϕ to fit the noise as:

$$\arg\min_{\phi} ||g_{\phi}(\mathbf{y}_1) - \mathbf{y}_2||_2^2. \tag{23}$$

Zeroshot-N2N [33] emphasizes that residual learning, a symmetry loss, and an additional coherence-enhancing term are critical for good performance. Zeroshot-N2N proposes two different loss functions, the residual loss $\mathcal{L}_{res}$ and the consistency loss $\mathcal{L}_{cons}$. First, the residual loss optimizes the network g_ϕ to fit the noise instead of image. The loss then becomes as:

$$\arg\min_{\phi} ||\mathbf{y}_1 - g_{\phi}(\mathbf{y}_1) - \mathbf{y}_2||_2^2. \tag{24}$$

To fit the noise in $\mathbf{y}_1$ and $\mathbf{y}_2$ both, a symmetric loss [34] is applied as:

$$\mathcal{L}_{res}(\phi) = \frac{1}{2}\big(||\mathbf{y}_1 - g_{\phi}(\mathbf{y}_1) - \mathbf{y}_2||_2^2 + ||\mathbf{y}_2 - g_{\phi}(\mathbf{y}_2) - \mathbf{y}_1||_2^2\big). \tag{25}$$

Second, the method constrain consistency by making denoised output of the downsampled image and downsampled result of the denoised image like:

$$\arg\min_{\phi} ||\mathbf{y}_1 - g_{\phi}(\mathbf{y}_1) - D_1(\mathbf{y}_1 - g_{\phi}(\mathbf{y}_1))||_2^2. \tag{26}$$

Same as Eq. 25, with the adoption of a symmetric manner, the consistency loss is represented as:

$$\mathcal{L}_{cons}(\phi) = \frac{1}{2}\big(||\mathbf{y}_1 - g_{\phi}(\mathbf{y}_1) - D_1(\mathbf{y}_1 - g_{\phi}(\mathbf{y}_1))||_2^2 + ||\mathbf{y}_2 - g_{\phi}(\mathbf{y}_2) - D_2(\mathbf{y}_2 - g_{\phi}(\mathbf{y}_2))||_2^2\big). \tag{27}$$

The noise removal loss function $\mathcal{L}_{noise}$ in Zeroshot-N2N becomes the sum of Eq. 25 and Eq. 27, expressed as:

$$\mathcal{L}_{noise} = \mathcal{L}_{res} + \mathcal{L}_{cons}. \tag{28}$$

A.4.3 More Quantitative Results

We present the comparison results for non-reference metrics, which we did not include in Table 1. Table 7 demonstrates that CLODE outperforms other unsupervised methods in terms of perceptual quality. Notably, it demonstrates competitive results in terms of BRISQUE and PI, even when compared to state-of-the-art supervised methods.

Table 7: Comparison results on LSRW [4] and LOL [5] in terms of NIQE [44], BRISQUE [45], PI [46] and Entropy [47]. Within the unsupervised approaches, the best score is displayed in **Red**. LLNODE performs better than all other methods, including supervised methods, in terms of PI (Perceptual Index).

Training	Method	LSRW				LOL			
		NIQE↓	BRISQUE↓	PI↓	Entropy↑	NIQE↓	BRISQUE↓	PI↓	Entropy↑
Supervised	Afifi et al. [49]	6.655	46.645	6.470	7.065	4.966	33.546	5.741	7.173
	RetinexNet [5]	-	-	-	-	8.871	51.813	4.955	6.835
	URetinexNet [37]	4.154	23.614	3.495	6.762	4.250	15.633	3.372	6.926
	LLFlow [1]	3.756	26.671	3.176	7.369	5.709	35.022	4.530	7.141
	RetinexFormer [2]	3.549	15.951	3.208	7.131	3.478	17.101	3.102	7.074
Unsupervised	SCI-easy [11]	3.847	25.859	3.259	6.388	7.153	12.424	5.437	5.825
	SCI-medium [11]	3.917	22.416	3.159	6.494	7.861	25.870	4.583	6.842
	SCI-difficult [11]	4.368	20.692	3.851	5.975	8.060	26.823	4.664	6.675
	RUAS [10]	5.426	38.854	4.939	6.442	6.303	11.977	4.571	7.194
	ZeroDCE [6]	3.776	23.867	3.156	6.526	7.777	27.301	4.459	6.608
	Night-Enhancement [17]	7.208	51.356	6.801	6.544	4.491	27.122	4.436	**7.139**
	PairLIE [13]	**3.684**	29.816	3.426	6.923	**4.083**	20.592	3.052	6.823
	CLODE	3.827	**18.426**	**3.115**	**7.025**	4.516	**8.220**	**2.914**	7.053

A.4.4 Comparison with other iterative methods

Fig. 9 shows the changes in performance over steps of each curve-adjustment-based method. Each comparison method is retrained for 10 steps in the official code provided by the author. To fix the number of steps in CLODE to 10, we replace CLODE's ODE solver with the Euler method, and referred to it as CLODE-*Euler*. The results show that even within the same number of steps, CLODE-*Euler* performs better than other curve adjustment-based methods. Furthermore, the proposed version, CLODE, demonstrates higher performance compared to other methods in most iterative steps.

In case of ReLLIE [9], it exhibits a decline in performance after 7 steps, emphasizing the need for careful selection of the number of iterative steps itself to achieve optimal result, this makes the method impractical to use.

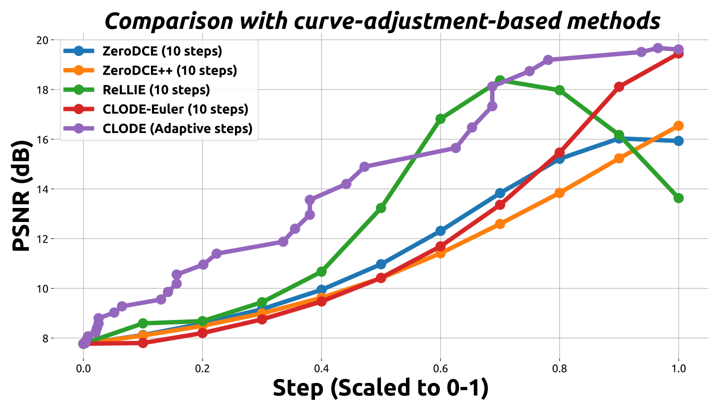


Figure 9: Changes in PSNR (Peak Signal-to-Noise Ratio) over steps of CLODE, CLODE-*Euler*, ReLLIE [9], ZeroDCE++ [8], and ZeroDCE [6]. As CLODE employs a continuous adaptive step according to the input image, we represent the steps by scaling them from 0 to 1. CLODE demonstrates superior performance compared to other methods at almost every step.

A.5 More visual results

We show additional results for CLODE enhancement that we did not show in the main manuscript due to lack of space. We present additional visual comparison results for PairLIE [13] and Night-Enhancement [17], which demonstrated the best quantitative performance among the unsupervised methods in Table 1 of the main manuscript, except for our proposed method (CLODE), in Fig. 10. CLODE shows the most robust enhancement results across various image exposure conditions.

Fig. 11, Fig. 12, Fig. 13 and Fig. 14 show the results for CLODE and CLODE† on LOL [5] and SICE [35] validation dataset. Additionally, Fig. 15 shows the visual results with different exposures for photos extracted from MSEC [49] and the internet (Filckr: CC BY-NC 2.0).

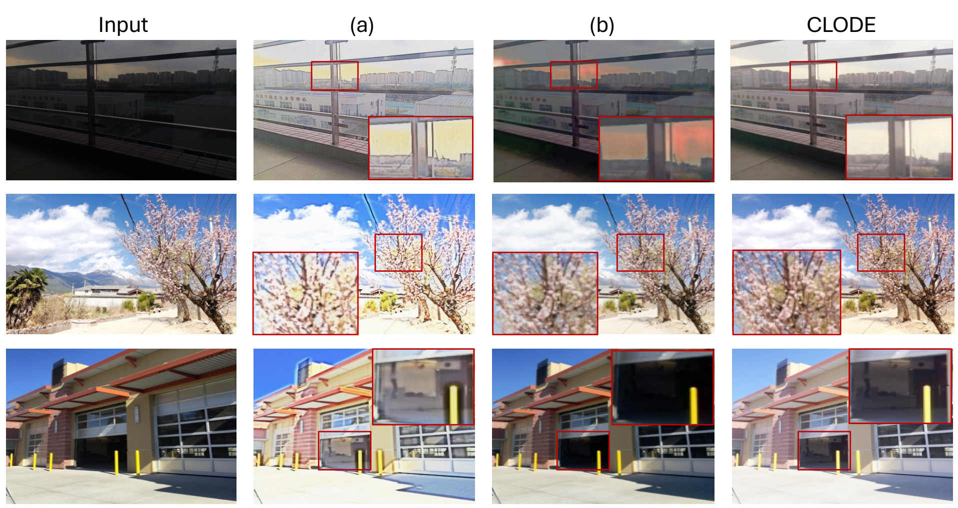


Figure 10: Comparative visualization results with (a) PairLIE [13] and (b) night-enhancement [17] on LOL [5] and SICE [35]. Images are taken from LSRW [4] and SICE [35] Part2.

Figure 11: Visualization results on LOL [5]. While CLODE demonstrates superior enhancement results, user control with CLODE† produces images that more closely resemble the ground-truth image.

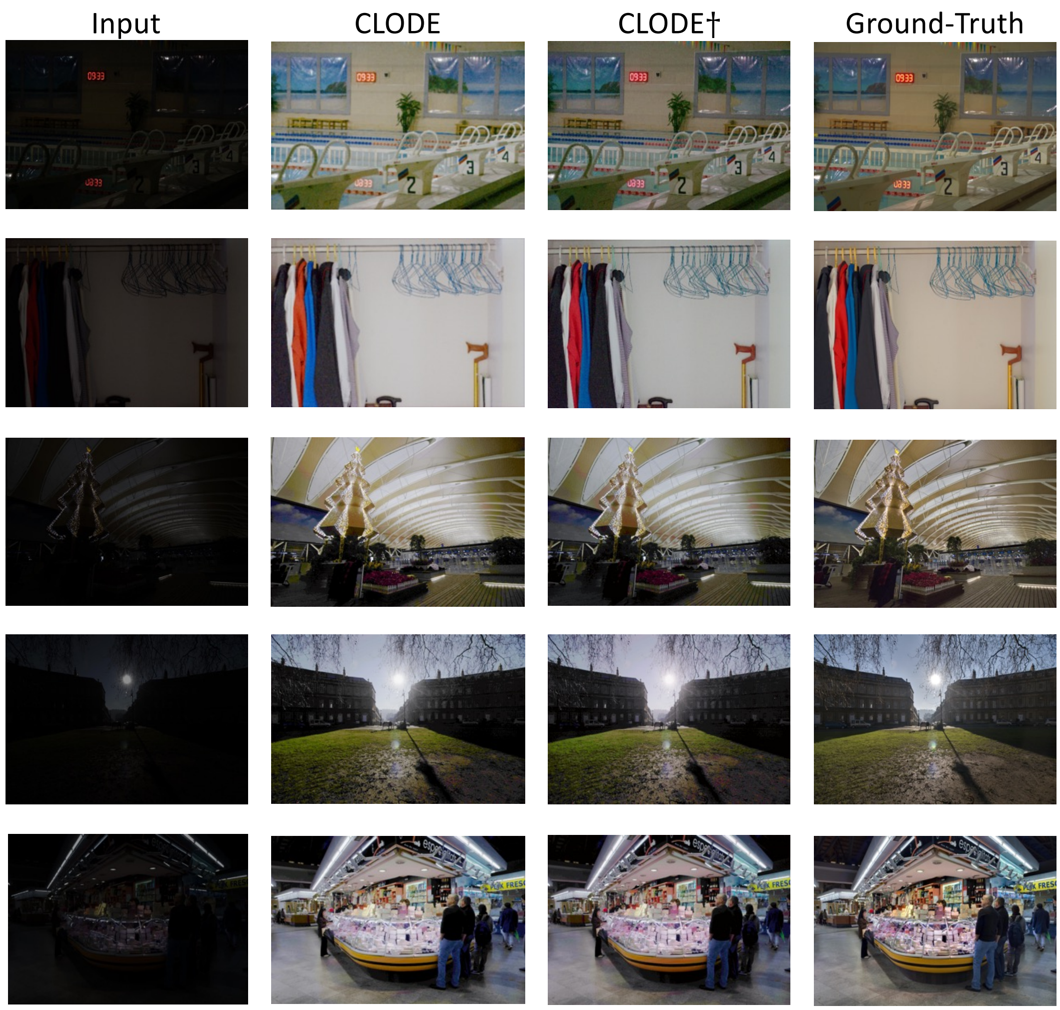


Figure 12: Visualization results on LOL [5] and SICE [35]. While CLODE demonstrates superior enhancement results, user control with CLODE† produces images that more closely resemble the ground-truth image.

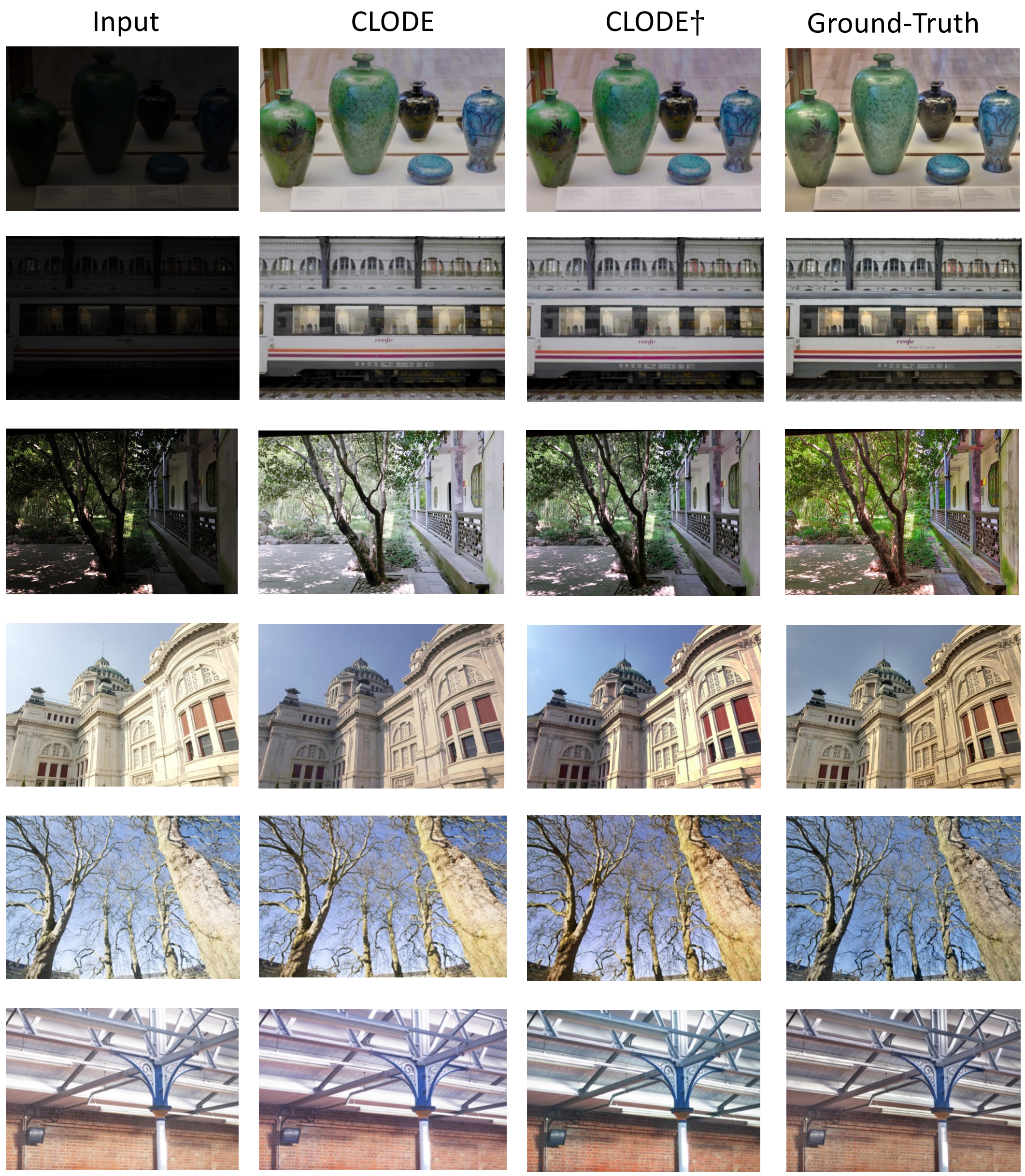


Figure 13: Visualization results on SICE [35]. While CLODE demonstrates superior enhancement results, user control with CLODE† produces images that more closely resemble the ground-truth image.

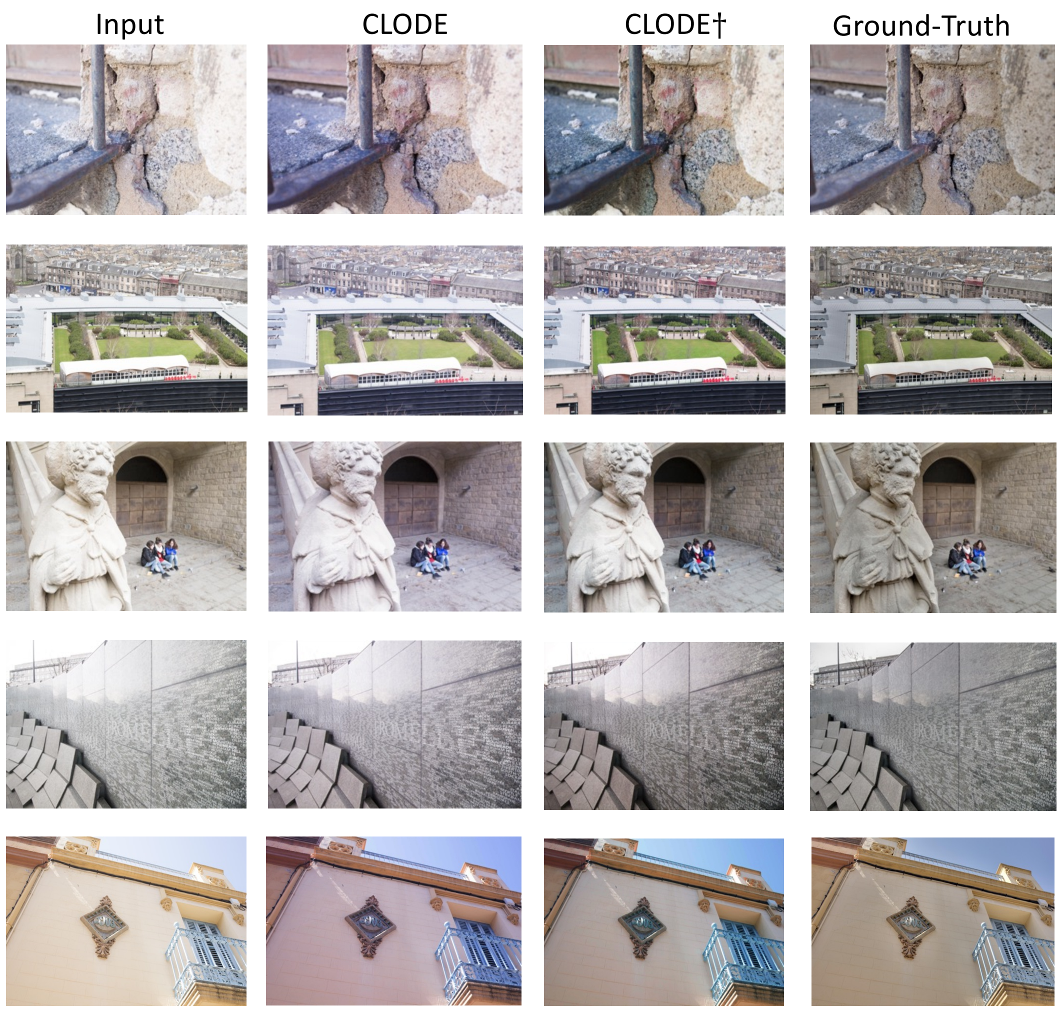


Figure 14: Visualization results on SICE [35]. While CLODE demonstrates superior enhancement results, user control with CLODE† produces images that more closely resemble the ground-truth image.

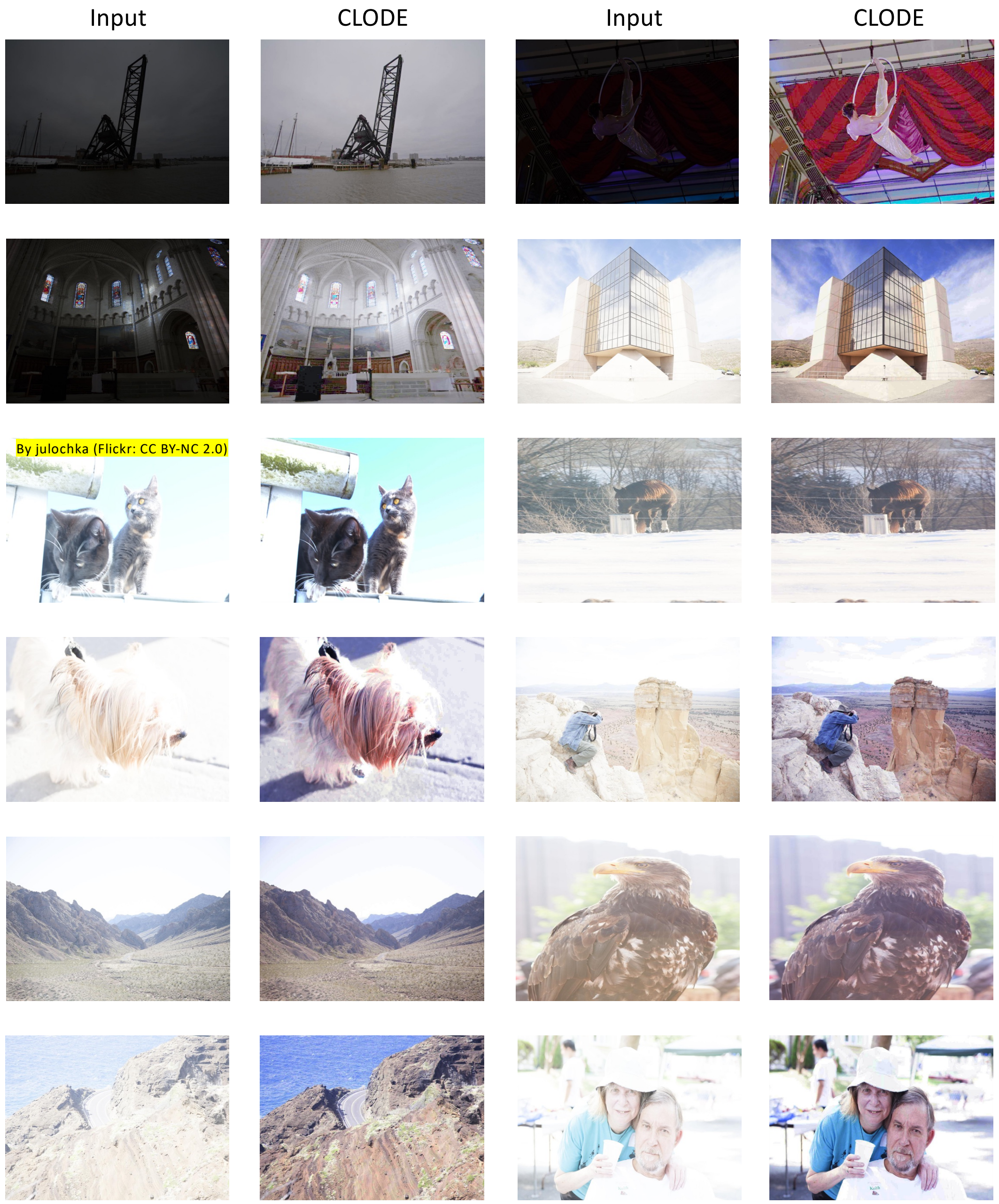


Figure 15: Visualization results on MSEC [49] and extracted from internet (Flickr by julochka). Even with diverse inputs of various exposures, CLODE show robust result in an unsupervised manner.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We presented the contributions and effects of our method in the abstract and introduction, and demonstrated them through experiments.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: There is a "Limitation" section containing information about the execution speed of our method.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Our paper reports the theoretical approaches of the NODE reformulation process for curve adjustment equations.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Our paper includes main experimental results as well as ablation experiment results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: In our paper, we include the rationale behind NODE reformulation along with the workflow, and provide the network architecture in the appendix. This ensures reproducibility, and we will also provide the code upon acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We included details about the dataset used for training in the main manuscript, while other hyperparameters, weights, etc., are documented in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide detailed information on low-light performance validation, the impact of main contributions, and performance validation through ablation studies.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide information about the GPU resources used and the execution time.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the ethical guidelines (Code of Ethics) and ensured compliance.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of our work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our proposed model does not include a high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We will accurately specify the original owner's license and share the code upon acceptance. Additionally, the source and the original owner's name for the single image used in the appendix have been indicated on the image.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not introduce new assets in this paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.