Pattern Recognition 156 (2024) 110799

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

# Denoising diffusion post-processing for low-light image enhancement

Savvas Panagiotou *, Anna S. Bosman

*Department of Computer Science, University of Pretoria, South Africa*

ARTICLE INFO

*Keywords:*
Diffusion model
Denoising
Low-light image enhancement
Post-processing

ABSTRACT

Low-light image enhancement (LLIE) techniques attempt to increase the visibility of images captured in low-light scenarios. However, as a result of enhancement, a variety of image degradations such as noise and color bias are revealed. Furthermore, each particular LLIE approach may introduce a different form of flaw within its enhanced results. To combat these image degradations, post-processing denoisers have widely been used, which often yield oversmoothed results lacking detail. We propose using a diffusion model as a post-processing approach, and we introduce Low-light Post-processing Diffusion Model (LPDM) in order to model the conditional distribution between under-exposed and normally-exposed images. We apply LPDM in a manner which avoids the computationally expensive generative reverse process of typical diffusion models, and post-process images in one pass through LPDM. Extensive experiments demonstrate that our approach outperforms competing post-processing denoisers by increasing the perceptual quality of enhanced low-light images on a variety of challenging low-light datasets. Source code is available at https://github.com/savvaki/LPDM.

## 1. Introduction

The task of low-light image enhancement (LLIE) aims to improve the visibility of images which are captured under low-light conditions. Images which are under-exposed to light are often degraded in a variety of ways in addition to their lack of visibility. Notably, low-light regions of an image typically contain degraded color information, a lack of detail as well as intensive noise. LLIE techniques aim to brighten low-light regions of an image while maintaining color accuracy and minimizing noise. The demand for brightening and enhancing low-light images often arises due to many downstream algorithms only being performant on images with high visibility [1]. Some of these downstream tasks include object detection [2], facial recognition [3], surveillance [4] and semantic segmentation [5].

Simply adjusting the contrast of low-light images using a technique such as histogram equalization [6] is often insufficient due to the amplification of noise [1,7]. Learning-based methods have emerged which significantly outperform traditional methods. However, even the state-of-the-art deep learning (DL) techniques still introduce a variety of artifacts in different scenarios [8].

Existing denoising techniques can be applied to denoise low-light images either before or after contrast enhancement [9,10]. These denoising techniques range from low-pass filters and algorithms such as block matching and 3D filtering (BM3D) [11], to state-of-the-art DL denoisers [9,12,13]. Despite existing denoisers significantly reducing noise, they often oversmooth the output. As a result, removing the amplified noise in a brightened low-light image comes at the cost of removing detail, especially in high-frequency regions of the image.

Recently, diffusion models (DMs) have emerged as a class of generative models with the ability to model complex conditional data distributions [14,15]. DMs are typically used to generate high-quality samples from a learned distribution by iteratively refining a random noise sample using the reverse diffusion process. DMs outperform other generative modeling paradigms in terms of their produced sample quality [16]. Therefore, considering that alternative generative models have been proven to be effective for LLIE in past research [17,18], the application of DMs in the LLIE domain presents a promising avenue for further investigation.

In this work, we propose a post-processing conditional DM [15] with the capability of removing unwanted noise and other distortions in brightened low-light images. We name our conditional model **L**ow-light **P**ost-processing **D**iffusion **M**odel (LPDM). The effect of post-processing using LPDM is displayed in Fig. 1. LPDM learns the conditional distribution between low-light and normally-exposed images. Thereafter, LPDM is applied as a post-processing step to remove noise and improve the sharpness and color accuracy of the enhanced low-light image. We introduce a novel approach of applying LPDM which allows the model to avoid the computationally expensive reverse diffusion process of typical DMs, and denoise a given image in one pass through the model. In summary, our contributions are as follows:

* Corresponding author.
*E-mail addresses:* u17215286@tuks.co.za (S. Panagiotou), anna.bosman@up.ac.za (A.S. Bosman).

https://doi.org/10.1016/j.patcog.2024.110799
Received 10 July 2023; Received in revised form 15 January 2024; Accepted 15 July 2024
Available online 19 July 2024

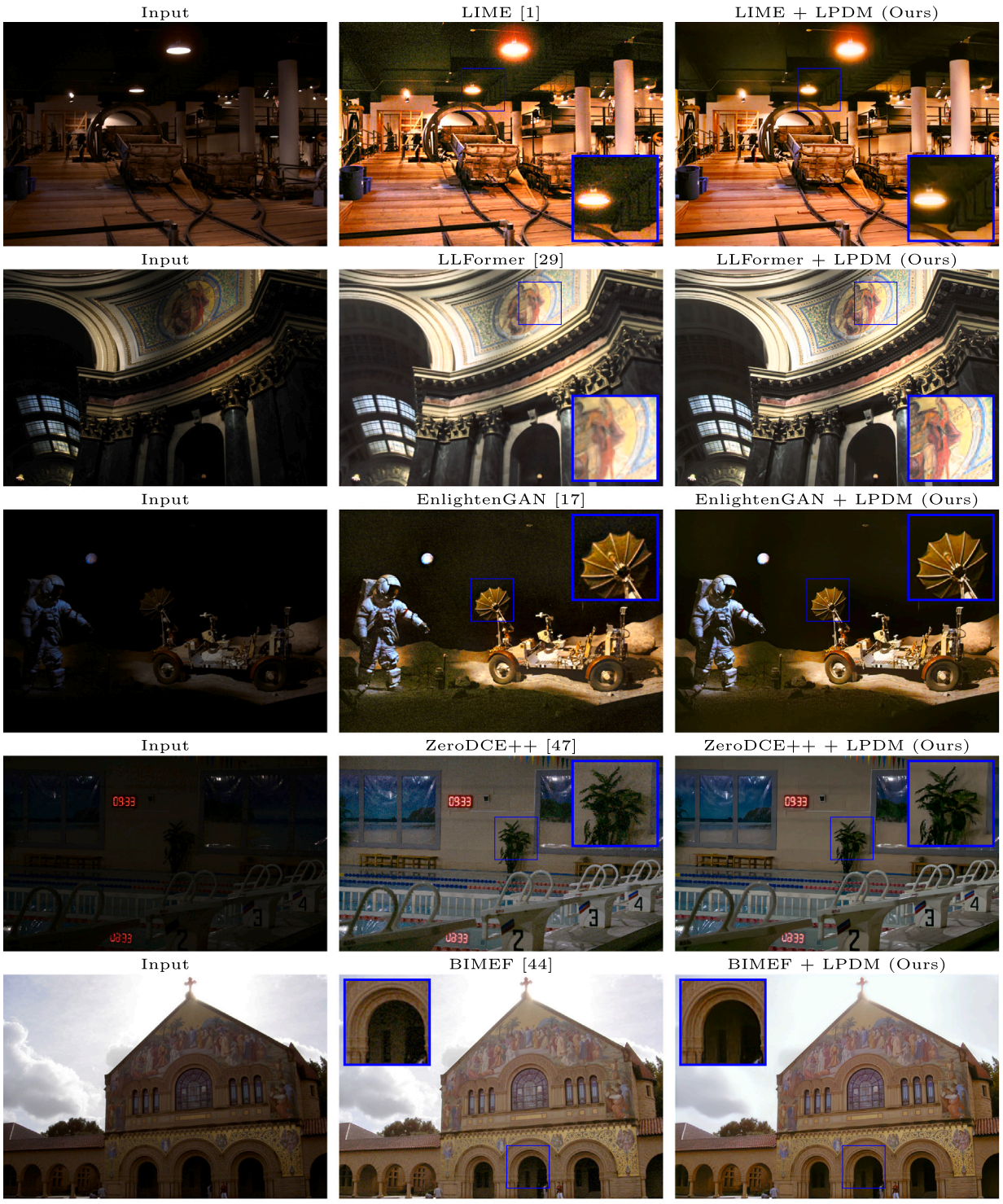


**Fig. 1.** Qualitative results of the proposed approach. Each respective column represents $c$, $x_0^{\eta}$ and $x_0^{DM}$.

1. We introduce a novel method of applying DMs as a post-processing technique in the LLIE pipeline. Our framework is able to circumvent the computationally expensive iterative reverse process of DMs and denoise images in one pass through the model without iteration.
2. We demonstrate that our DM improves existing state-of-the-art LLIE techniques on popular low-light datasets including challenging unpaired test sets.
3. In addition to simple denoising, we demonstrate that our method is able to cope with a variety of different artifacts and color distortions, yielding superior results to existing post-processing denoisers for LLIE.

The remainder of this paper is structured as follows: Section 2 provides background information on LLIE; Section 3 describes existing post-processing denoisers and their limitations; Section 4 provides background information on DMs; Section 5 outlines preliminary mathematical notation and describes the proposed framework in detail; Section 6 contains the experimental setup and results for this work, including an ablation study; finally, conclusions are drawn in Section 7.

## 2. Low-light image enhancement

LLIE techniques have existed for many decades and can be divided into non-learning-based methods and learning-based methods. Popular examples of traditional techniques which do not require learning from data include variants of histogram equalization (HE) [6,19] and gamma correction (GC) [20]. HE adjusts the global contrast of an image via a single transformation function. However, low-light images often require contrast enhancements that vary dynamically depending on local regions of the image. Thus, techniques such as GC adjust an image via a non-linear per-pixel transform to brighten dark regions while leaving bright regions relatively unaffected. Despite achieving reasonable results, the abovementioned traditional methods often require post-processing techniques in order to deal with amplified noise after enhancement, and struggle to perform well across diverse scenes.

Another paradigm of LLIE makes use of Retinex theory [21], where the assumption is that a color image can be separated into reflectance and illumination, such that $\mathbf{L} = \mathbf{R} \circ \mathbf{T}$, where $\mathbf{L}$ represents the source low-light image, $\mathbf{R}$ represents the desired recovery image (reflectance), $\mathbf{T}$ represents the illumination and $\circ$ represents element-wise multiplication. Non-learning Retinex-based methods such as Low-light Image Enhancement via Illumination Map Estimation (LIME) [1] provide an effective image enhancement approach by estimating $\mathbf{T}$, which is then used to determine $\mathbf{R}$ given $\mathbf{L}$; however, post-processing denoising is typically still necessary using algorithms such as BM3D [11] which often blurs high-frequency details. Alternative Retinex-based methods reformulate the traditional Retinex model to incorporate an added noise term in order to cater for noise [22].

DL methods have recently achieved state-of-the-art LLIE performance. Some complexities of catering for a large variety of realistic low-light scenes are abstracted away by a data-driven approach. Most DL architectures are based on Convolutional Neural Networks (CNNs) and more recently, CNNs have been hybridized with transformer networks [23]. DL methods either opt for incorporating denoising into a single model, or apply denoising as a post-processing step. S-LLNet [4] makes use of a learned denoiser which operates sequentially after contrast enhancement. Retinex-net [24] incorporates Retinex theory into a DL model with an optional illumination-varying BM3D denoiser used as post-processing. Raw sensor data are enhanced and denoised via CNN in [7]. DCC-Net [25] aims at enhancing illumination whilst preserving color information. MSINet [26] enhances low-light stereo image pairs. A variety of loss functions have also been proposed (in addition to the typical $l_1$ and $l_2$ losses) which further penalize networks based on color, smoothness, brightness and perceptual interpretation [27]. Unsupervised Generative Adversarial Networks (GANs) have also been proposed for LLIE [17]. Recently, transformer architectures have gained popularity for LLIE which exploit spatial and channel-wise attention mechanisms [28,29].

Two more relevant state-of-the-art approaches with respect to this work are the DL models LLFlow [18] and GSAD [30]. The LLFlow framework learns the conditional distribution between low-light and normally-exposed images via the generative paradigm of normalizing flow [31]. GSAD [30] applies diffusion models in order to enhance low-light images iteratively via conditioning on the low-light image.

LPDM proposed in this study also models the conditional distribution between low-light and normally-exposed images using the diffusion paradigm. Employing a DM allows our technique to learn non-linear complexities of the conditional distribution that may elude alternative models. Furthermore, in our novel approach, we repurpose the function of a DM to be used as a noise extractor rather than a generative image sampler. Therefore, LPDM provides a subtractable estimation of the noise in an image which can further enhance the image without iteration. In contrast to LLFlow and GSAD, LPDM is used as a post-processing step which can be applied regardless of the enhancing step that precedes LPDM. To the best of the authors' knowledge, LPDM is the first application of denoising DMs in the field of post-processing LLIE.

## 3. Post-processing denoising

The task of LLIE can partially be considered a denoising problem, since brightening the image is only one component of the task, with the other component being noise suppression. Existing LLIE methods often fail to suppress degradations in their output [27]. Therefore, post-processing denoising approaches have been used to handle noise after enhancement. In this work, the denoising performance of LPDM is compared with that of the BM3D algorithm [11], and two state-of-the-art DL denoisers: NAFNet [12] and MIRNet* [13].

BM3D is a popular choice of post-processing denoiser in the LLIE field, which has been used by RetinexNet and LIME. Naively applying BM3D evenly over the entire image results in bright regions being oversmoothed, since noise is often present at higher levels in dark regions. Therefore, previous LLIE works have adapted BM3D to operate unevenly based on the illumination of the recovered image [1,24]. In particular, we compare our approach to the BM3D denoising approach used in LIME [1].

Secondly, we compare our approach to NAFNet, which is a state-of-the-art supervised U-Net model that avoids the use of non-linear activation functions [12,32]. NAFNet is trained on the smartphone image denoising dataset (SIDD) [33]. SIDD contains pairs of clean and noisy images captured under multiple lighting conditions. Therefore, NAFNet is well-suited to denoise enhanced images of a variety of brightness levels. Finally, we compare our LPDM to another state-of-the-art denoiser MIRNet*, which is a MIRNet model [34] that is fine-tuned using realistic noise generated by PNGAN [13]. MIRNet* is trained on the SIDD dataset, as well as additional datasets that have synthetic noise added by PNGAN.

## 4. Diffusion models

A DM is a form of generative model which has recently been shown to generate high-quality samples, outperforming GANs [14–16,35]. DMs iteratively remove small perturbations of noise, typically starting with a sample from an isotropic Gaussian distribution, until they generate a clean data sample. In this way, the unconditional diffusion process connects a complex data distribution $q(\boldsymbol{x}_0)$ to a simpler, analytically tractable distribution via a Markov chain consisting of a finite number of timesteps $T$ [14]. The subscript of a sample indicates a timestep in the Markov chain, with 0 being a clean sample and $T$ being a sample with the maximum amount of noise added. DMs have been successfully used to model both unconditional and conditional distributions [35,36]. In spite of the impressive results of DMs, the speed of generating samples has always been a drawback due to their iterative reverse process. Attempts have been made to increase sample speed by making the sampling process non-Markovian, as well as moving DMs to the latent space [37,38].

We avoid using a DM for sampling normally-exposed images owing to the expensive generative reverse process. Instead, we exploit the ability of DMs to capture complex conditional data distributions. In particular, we use a DM to capture the relationship between under-exposed and normally-exposed images. Other work has shown that DMs may be used as backbone feature extractors which predict features based on noisy inputs [39,40]. Similarly, our work exploits a DM's ability to identify anomalies which are outside the conditional distribution between normally-exposed and under-exposed images. The applications of using DMs in the field of LLIE are relatively unexplored, especially considering that LLIE can be posed as a denoising problem.

## 5. Methodology

In this work, we propose a technique where a conditional DM is used to remove noise from images which have undergone LLIE. The remainder of this section is structured as follows: in Section 5.1, the background information about DMs is outlined; the architecture used for LPDM is described in Section 5.2; finally, in Section 5.3, we provide detail of our proposed framework.

### 5.1. Preliminaries

DMs make use of a forward process which adds noise to a sample and a reverse process which removes noise. The objective is to model the conditional data distribution $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0|\boldsymbol{c})$, where $\boldsymbol{c}$ is a conditioning image and $\boldsymbol{x}_0$ is a target image. The forward diffusion process is defined as follows [15]:

$$\begin{aligned} q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) &:= \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}), \\ q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) &:= \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t\mathbf{I}), \end{aligned} \tag{1}$$

where $\beta_t \in (0,1)$ defines a variance to be used at timestep $t$. As seen by the Markov chain in Eq. (1), obtaining a more noisy sample $\boldsymbol{x}_t$ is dependent on the previous less-noisy sample $\boldsymbol{x}_{t-1}$. Extra notation is defined such that $\alpha_t := 1-\beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$, which allows for Eq. (1) to be reformulated to be conditioned on the original clean data sample $\boldsymbol{x}_0$ [15]:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) := \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I}). \tag{2}$$

The efficient sampling of an arbitrary $\boldsymbol{x}_t$ at any timestep in the Markov chain is possible given $\boldsymbol{x}_0$:

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \tag{3}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0,1)$ is a random source. The variance schedule is designed such that $\boldsymbol{x}_T \approx \mathcal{N}(0,1)$. Conditional DMs model the reverse process $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c})$ where $\theta$ indicates that the DM is modeled by a neural network parameterized by $\theta$. The conditional DM attempts to maximize the likelihood $p_\theta(\boldsymbol{x}_0|\boldsymbol{c})$. The reverse diffusion process is defined by parameterized Gaussian transitions [15]:

$$\begin{aligned} p_\theta(\boldsymbol{x}_{0:T}|\boldsymbol{c}) &:= p(\boldsymbol{x}_T) \prod_{t=1}^{T} p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c}), \\ p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c}) &:= \mathcal{N}(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t, \boldsymbol{c}), \boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t, \boldsymbol{c})). \end{aligned} \tag{4}$$

In order to avoid learning the variance, let $\boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t, \boldsymbol{c}) = \sigma_t^2\mathbf{I}$, where $\sigma_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$ is a time-dependent constant [15]. Therefore, the only learnable component is $\boldsymbol{\mu}_\theta$. Instead of directly predicting $\boldsymbol{\mu}_\theta$, the DM is parameterized in terms of a denoising autoencoder $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t, \boldsymbol{c})$ where $t = 1, \ldots, T$. The number of timesteps $T$ is set to a large number (such as $T = 1000$) in order for the reverse process to better-approximate a Gaussian distribution [37]. The corresponding simplified objective is as follows [15]:

$$L_{DM} = \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{c}, \boldsymbol{\epsilon}, t}[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}), t, \boldsymbol{c}\|^2], \tag{5}$$

where $t$ is uniformly sampled from $\{1, \ldots, T\}$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(0,1)$. In simplified terms, $L_{DM}$ guides the DM to predict the underlying $\boldsymbol{\epsilon}$ that was involved in sampling $\boldsymbol{x}_t$. Given $\boldsymbol{x}_t$ and a prediction for $\boldsymbol{\epsilon}$ using $\boldsymbol{\epsilon}_\theta$, we can calculate an estimate of $\boldsymbol{x}_0$ [15]:

$$\boldsymbol{x}_0 \approx \hat{\boldsymbol{x}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\boldsymbol{x}_t - \left(\sqrt{\frac{1}{\bar{\alpha}_t} - 1}\right)\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t, \boldsymbol{c}). \tag{6}$$

Further information about the DM sampling process is omitted since it is not used in this work.

### 5.2. Diffusion model architecture

The DM architecture $\boldsymbol{\epsilon}_\theta$ used for modeling the diffusion process is typically a form of modified U-Net [15,32]. For the purpose of creating a noise-prediction model, the original U-Net has been improved since its inception. The popularized DM architecture is based on a PixelCNN++ backbone which incorporates residual blocks into the U-Net [15,41]. Furthermore, the DM operates across all timesteps in the Markov chain using the same parameters. This is made possible by modifying the original U-Net to condition on timestep information. The integer timestep $t$ is converted to a continuous vector embedding using a sinusoidal positional embedding strategy [23]. The residual blocks within the U-Net therefore accept timestep embedding input in addition to the input feature maps, and thus each residual block has context as to what level of noise should be present in the input features.

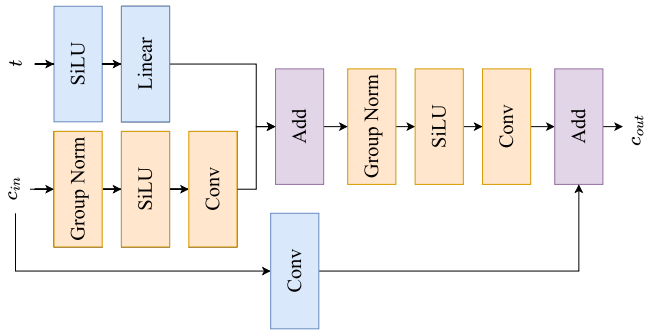


**Fig. 2.** Residual block used throughout the DM architecture consisting of a combination of group normalization layers, SiLU activations, convolution layers and addition operations. Both $c_{in}$ and $t$ are inputs to the residual block and represent the channel and timestep-embedded input respectively. Note that $t$ is already in embedded form when it enters the residual block. The output of the residual block is represented by $c_{out}$.

Other notable modifications from the original U-Net are the use of attention mechanisms at different spatial resolutions as well as the use of group normalization [42] within the residual blocks. As shown in Fig. 2, the residual blocks of the DM are composed of a combination of group normalization layers, SiLU activations [43], convolutional layers and addition operators. A variety of attention mechanisms may be applied at different spatial resolutions, such as scaled dot-product attention, for example [23].

### 5.3. Proposed framework

In order to address possible degradations which occur after LLIE, we propose LPDM, represented by $\boldsymbol{\epsilon}_\theta$. The LPDM model is given $(\boldsymbol{x}_t \oplus \boldsymbol{c}, t)$ as input and tasked with predicting $\boldsymbol{\epsilon}$, where $\boldsymbol{x}_t \sim q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ is a normally-exposed image with noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0,1)$ added at a timestep $t$, $\boldsymbol{c}$ is the corresponding under-exposed image and $\oplus$ is the concatenation operator. During training, LPDM is exposed to batches of $(\boldsymbol{x}_t \oplus \boldsymbol{c}, t)$ with randomly sampled $t$ for each sample in a batch. A visualization of the training process is depicted in the left half of Fig. 3.

After LPDM has been trained, LPDM is applied in a novel manner different to that of conventional DMs. Current DMs generate samples by applying the DM iteratively to make noise predictions for each timestep (depending on the sampling strategy), as shown in the top half of Fig. 4. The maximum number of iterations required is $T$, which is often set to 1000, making the sampling process prohibitively slow. In contrast, our approach only requires one prediction from LPDM, shown in the bottom half of Fig. 4, which makes our approach more computationally efficient than the conventional iterative sampling approach. In order to achieve this ability, LPDM assumes that its input is already somewhat enhanced. Then, LPDM compares the enhanced image to the low-light image conditioning and predicts flaws in the enhanced image as noise. This noise prediction is then subtracted from the enhanced image to repair it without iteratively using LPDM. The proposed approach is described in detail below.

Let $\eta$ be any low-light image enhancer and let $\hat{\boldsymbol{x}}_0^\eta$ be an enhanced image such that $\hat{\boldsymbol{x}}_0^\eta = \eta(\boldsymbol{c})$. LPDM is used to obtain an estimate of the noise present in $\hat{\boldsymbol{x}}_0^\eta$:

$$\boldsymbol{n}_\phi = \boldsymbol{\epsilon}_\theta(\hat{\boldsymbol{x}}_0^\eta, \phi, \boldsymbol{c}), \tag{7}$$

where $\phi < T$ is a timestep at which we wish to detect noise in $\hat{\boldsymbol{x}}_0^\eta$. An important property of Eq. (7) is that noise is not added to $\hat{\boldsymbol{x}}_0^\eta$; rather, the model is tasked with finding the noise present in $\hat{\boldsymbol{x}}_0^\eta$ as a result of enhancement, based on the conditioning $\boldsymbol{c}$ and timestep $\phi$. Additionally, the value of $\phi$ is selected at a level such that the underlying structure of the image would be preserved if noise were hypothetically to be added. Thus, a suitable value for $\phi$ is related to the

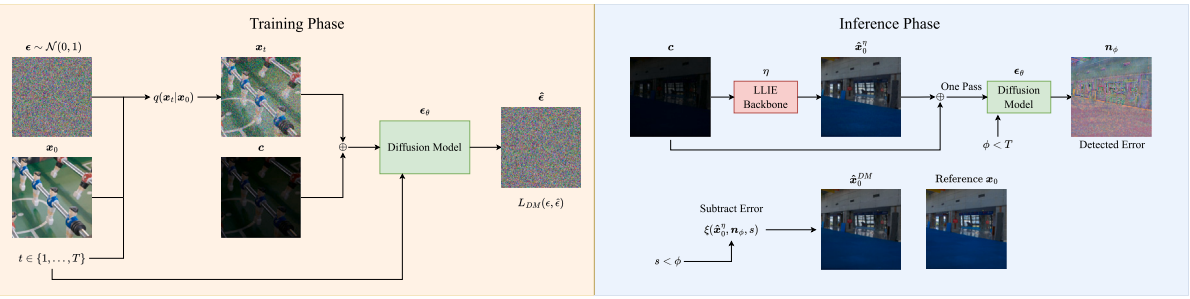


**Fig. 3.** Diagram presenting the training phase and inference phase of LPDM, displayed on the left and right half of the diagram respectively.

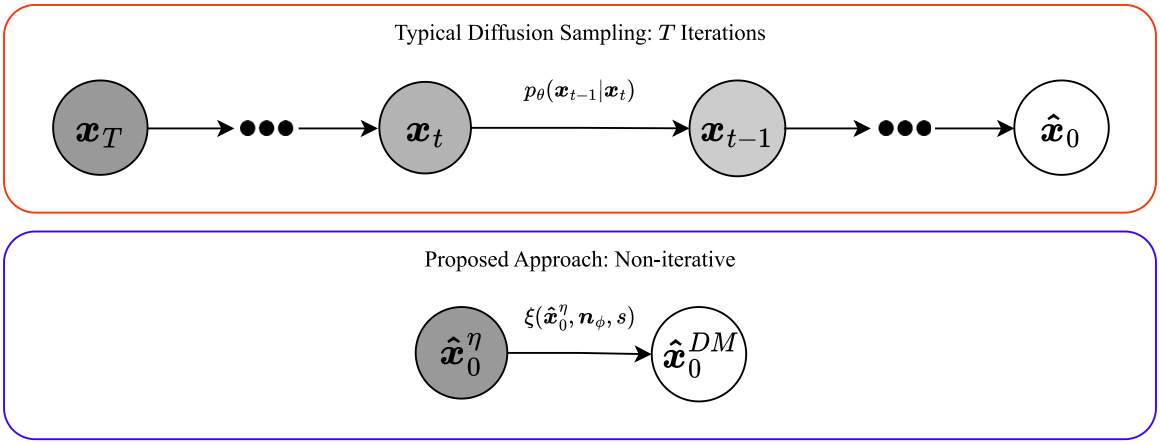


**Fig. 4.** Comparison between the typical diffusion sampling process and the proposed approach which does not require iteration.

selected variance schedule. Fig. 5 provides visual examples of applying our proposed approach for an example set of $\boldsymbol{x}_0$, $\boldsymbol{c}$ and $\hat{\boldsymbol{x}}_0^\eta$. Row 2 of Fig. 5 demonstrates the effect of selecting different values of $t$ in Eq. (2) in order to sample a noisy image during the training process. The noise schedule closer to 0 corresponds to less added noise in the image. Therefore, it is reasonable to detect noise in $\hat{\boldsymbol{x}}_0^\eta$ at lower levels of $\phi$ since we do not expect $\hat{\boldsymbol{x}}_0^\eta$ to be pure noise.

Row 3 of Fig. 5 demonstrates the effect of different values of $\phi$. For values of $\phi$ that are too low, the model overestimates the noise present in $\hat{\boldsymbol{x}}_0^\eta$. For large values of $\phi$, the model becomes similar to an autoencoder and attempts to predict the input. The reason for the behavior of different levels of $\phi$ can be explained by how LPDM is trained. For values of $\phi$ close to $T$, LPDM expects the input to be purely noise, and thus a suitable prediction for $\epsilon$ would simply be to predict the input. For low values of $\phi$, the model expects almost zero noise in the input. Therefore, the model produces an imperceptible granular estimation of the noise; however, this fails to consider the enhanced image distortions as noise. We find a good balance to be values of $\phi$ where the background structure of the image is not completely destroyed such as $\phi = 300$ which corresponds to $\boldsymbol{x}_{300}$ in row 2 of Fig. 5.

Once we obtain the estimation of the noise $\boldsymbol{n}_\phi$, we subtract the noise from $\hat{\boldsymbol{x}}_0^\eta$ using a modification of Eq. (6):

$$\hat{\boldsymbol{x}}_0^{DM} = \xi(\hat{\boldsymbol{x}}_0^\eta, \boldsymbol{n}_\phi, s) := \frac{1}{\sqrt{\bar{\alpha}_s}}\hat{\boldsymbol{x}}_0^\eta - \left(\sqrt{\frac{1}{\bar{\alpha}_s} - 1}\right)\boldsymbol{n}_\phi, \tag{8}$$

where $s$ is a timestep which selects the coefficients according to the variance schedule, and thus balances the degree to which noise is subtracted from $\hat{\boldsymbol{x}}_0^\eta$. The final result after LPDM post-processing is represented by $\hat{\boldsymbol{x}}_0^{DM}$. Notably, we find that $s$ should be significantly less than $\phi$ in order to subtract the correct amount of noise. As $s \to \phi$, more of the noise $\boldsymbol{n}_\phi$ is subtracted from $\hat{\boldsymbol{x}}_0^\eta$, leading to overcorrections and perhaps further degrading the result. As seen in row 4 of Fig. 5, the value of $s$ impacts how much correction should be applied. Our technique is able to reduce noise, correct color and improve sharpness as seen when comparing $\hat{\boldsymbol{x}}_0^\eta$ to $\xi(\hat{\boldsymbol{x}}_0^\eta, \boldsymbol{n}_{300}, 100)$ in Fig. 5.

Denoising techniques may be classified as being either blind or non-blind. Blind denoisers do not require the user to specify the level of noise in the input image, whereas non-blind denoisers require the user to specify the noise level. The popular BM3D algorithm is a non-blind approach. Similarly, our approach requires a selection of $s$ to determine to what extent noise should be subtracted, however we find low values of $s$ to be applicable to a wide variety of scenarios. Heuristics for determining optimal values of $s$ have been left for future research.

In summary, our approach requires the specification of a parameter $s$ during application, where $\phi$ may be fixed empirically. We find $\phi = 300$ to be a reasonable choice, and we use this value for all experiments. As $s \to 0$, the amount of correction lessens. The right half of Fig. 3 summarizes the inference process described above and shows that LPDM predicts $\hat{\epsilon}$ during training and $\boldsymbol{n}_\phi$ during inference.

## 6. Experiments

The following subsections outline the experimental setup: Section 6.1 describes the datasets used in this study; Section 6.2 defines the configuration of LPDM and the training parameters used for all experiments; Section 6.3 provides detail on the LLIE models selected for comparison with LPDM; in order to achieve a fair comparison, we compare our approach to alternative denoising methods described in Section 6.4; the interpretation of all the results is presented in Section 6.5; finally, an ablation study is conducted in Section 6.6.

### 6.1. Evaluation datasets

Paired low-light datasets are challenging to collect due to the requirement of having the scene remain unchanged while camera ISO is adjusted [24]. Accordingly, many methods resort to augmenting

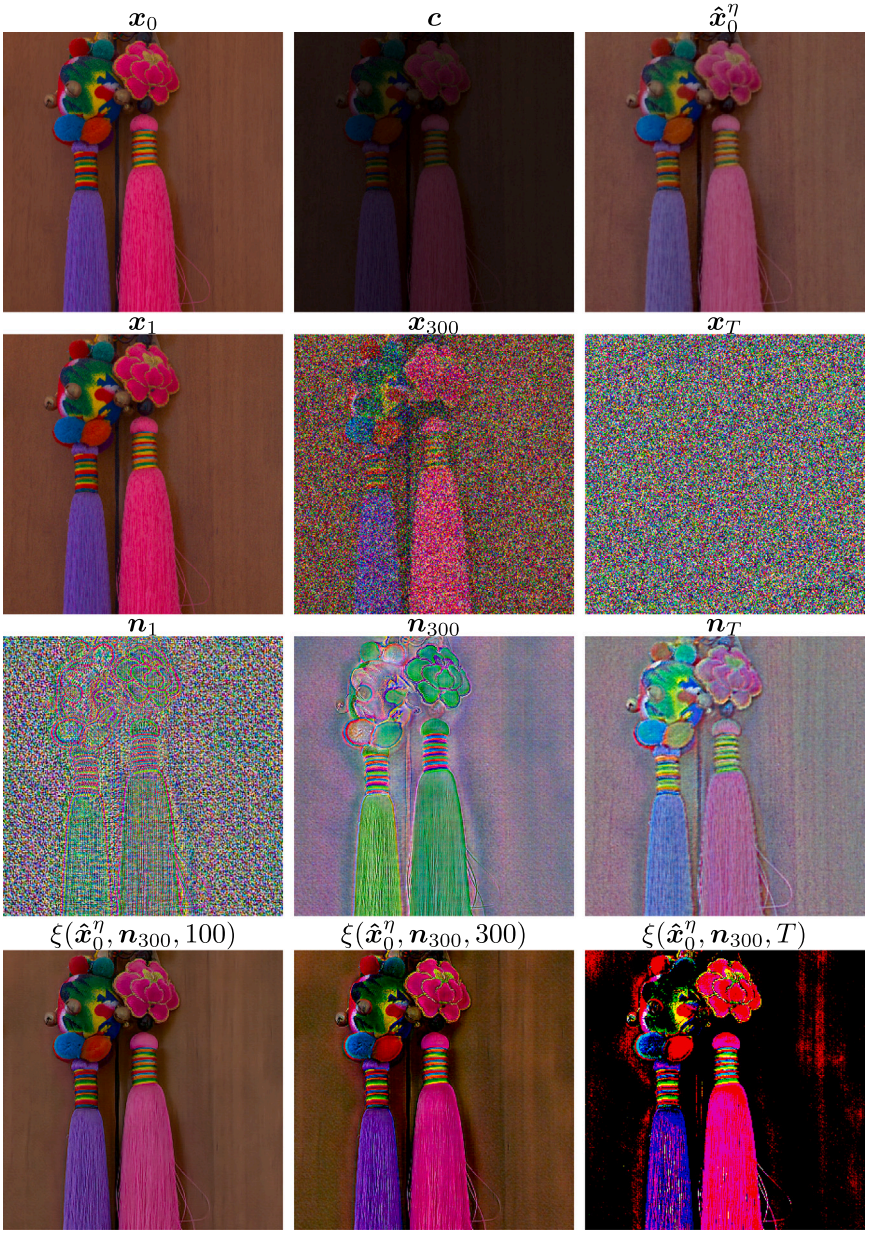


**Fig. 5.** Visualization of different components within the diffusion process and LPDM pipeline. The first row displays a normally-exposed image, under-exposed image and image which has undergone LLIE. The second row demonstrates how noise is added to $\boldsymbol{x}_0$ using a linear variance schedule during the training process, with $T = 1000$. The third row demonstrates the effect of different values of $\phi$ in Eq. (7). The fourth row demonstrates the effect of applying Eq. (8) with different values of $s$ in order to enhance $\hat{\boldsymbol{x}}_0^{\eta}$.

datasets with synthetic data. Synthetic datasets are typically generated by adjusting the gamma of normally-exposed images and adding simulated noise. We avoid training LPDM on synthetic datasets in order to ensure that we correctly model the conditional distribution between under-exposed and normally-exposed images. We train on the original paired LOL dataset [24], which contains 485 training images and 15 test images. Alternative versions of LOL exist, however these only add synthetic data such as the extended version of LOL [52] and VE-LOL [3]. We evaluate our model on the widely-adopted real unpaired test sets LIME[1] (10 images) [1], DICM (64 images) [49], MEF (17 images) [50], NPE (7 images) [51]. We specify the number of images explicitly as previous works use varying subsets of the test sets. The full-reference metrics we adopt are structural similarity index measure (SSIM), peak signal-to-noise ratio (PSNR), mean absolute error (MAE), Spatial-CIELAB ($\Delta E_s$) [53] and learned perceptual image patch similarity (LPIPS) [54]. For the unpaired test data we adopt the following no-reference metrics: natural image quality evaluator (NIQE) [55], blind/referenceless image spatial quality evaluator (BRISQUE) [56] and the MUSIQ transformer [57] trained on the smartphone photography attribute and quality (SPAQ) database [58]. All metrics are calculated in the RGB color space unless otherwise stated. The metric $\Delta E_s$ specifically measures perceptual color difference between two images. The output of the $\Delta E_s$ calculation is a spatial color error map, and the average of the color error map is reported.

### 6.2. Implementation details

A linear variance schedule is used in the range [0.00085, 0.012] for the diffusion process. The value of $T$ is fixed to 1000 for all experiments. LPDM is trained on the LOL training set for 6000 training steps using

[1] LIME [1] is both an LLIE technique and an unpaired test dataset, both of which are proposed in the same paper.

**Table 1**
Results on the LOL test set for different LLIE methods ($\eta$), with and without post-processing. The results for LPDM are averaged over 5 runs.

| Methods | SSIM ↑ | PSNR ↑ | MAE ↓ | $\Delta E_s$ ↓ | LPIPS ↓ | Methods | SSIM ↑ | PSNR ↑ | MAE ↓ | $\Delta E_s$ ↓ | LPIPS ↓ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LIME [1] | 0.484 | 17.181 | 0.124 | **17.159** | 0.372 | ZeroDCE [46] | 0.562 | 14.861 | 0.185 | **24.902** | 0.335 |
| LIME + $BM3D_5$ | 0.521 | 17.163 | 0.126 | 17.373 | 0.285 | ZeroDCE + $BM3D_5$ | 0.593 | 14.656 | 0.190 | 25.635 | 0.220 |
| LIME + $BM3D_{15}$ | 0.543 | 17.338 | 0.125 | 17.405 | 0.218 | ZeroDCE + $BM3D_{15}$ | 0.580 | 14.649 | 0.190 | 25.720 | 0.256 |
| LIME + NAFNet | 0.753 | **18.109** | **0.113** | 17.194 | 0.249 | ZeroDCE + NAFNet | 0.689 | 15.077 | 0.180 | 25.313 | 0.287 |
| LIME + MIRNet* | **0.764** | 18.107 | **0.113** | 17.270 | 0.257 | ZeroDCE + MIRNet* | **0.697** | **15.145** | **0.179** | 24.972 | 0.288 |
| LIME + $LPDM_{15}$ | 0.688 | 17.628 | 0.121 | 17.213 | 0.198 | ZeroDCE + $LPDM_{15}$ | 0.696 | 14.917 | 0.187 | 25.261 | **0.191** |
| LIME + $LPDM_{30}$ | 0.744 | 17.621 | 0.122 | 17.264 | **0.172** | ZeroDCE + $LPDM_{30}$ | 0.695 | 14.862 | 0.189 | 25.328 | **0.191** |
| BIMEF [44] | 0.595 | 13.875 | 0.206 | **29.698** | 0.326 | ZeroDCE++ [47] | 0.573 | 15.357 | 0.176 | 25.875 | 0.335 |
| BIMEF + $BM3D_5$ | 0.611 | 13.653 | 0.212 | 30.470 | 0.226 | ZeroDCE++ + $BM3D_5$ | 0.605 | 15.169 | 0.182 | 26.265 | 0.214 |
| BIMEF + $BM3D_{15}$ | 0.588 | 13.619 | 0.212 | 30.557 | 0.285 | ZeroDCE++ + $BM3D_{15}$ | 0.592 | 15.167 | 0.182 | 26.375 | 0.250 |
| BIMEF + NAFNet | 0.662 | 13.917 | **0.203** | 30.024 | 0.309 | ZeroDCE++ + NAFNet | 0.704 | 15.628 | **0.172** | 26.207 | 0.288 |
| BIMEF + MIRNet* | 0.669 | **13.946** | 0.204 | 29.916 | 0.313 | ZeroDCE++ + MIRNet* | **0.707** | **15.630** | **0.172** | 26.245 | 0.297 |
| BIMEF + $LPDM_{15}$ | **0.692** | 13.894 | 0.210 | 29.911 | **0.190** | ZeroDCE++ + $LPDM_{15}$ | **0.707** | 15.455 | 0.179 | 25.716 | **0.188** |
| BIMEF + $LPDM_{30}$ | 0.684 | 13.844 | 0.212 | 29.964 | 0.197 | ZeroDCE++ + $LPDM_{30}$ | 0.706 | 15.384 | 0.180 | **25.640** | 0.190 |
| RetinexNet [24] | 0.425 | 16.774 | 0.126 | 20.368 | 0.474 | LLFlow [18] | **0.852** | 21.133 | **0.084** | 13.125 | **0.119** |
| RetinexNet + $BM3D_5$ | 0.456 | 16.794 | 0.126 | 20.404 | 0.409 | LLFlow + $BM3D_5$ | 0.845 | 21.011 | 0.085 | 13.292 | 0.144 |
| RetinexNet + $BM3D_{15}$ | 0.482 | 17.065 | 0.123 | 19.913 | 0.278 | LLFlow + $BM3D_{15}$ | 0.830 | 20.937 | 0.086 | 13.314 | 0.189 |
| RetinexNet + NAFNet | 0.724 | 17.636 | 0.118 | 20.067 | 0.312 | LLFlow + NAFNet | 0.808 | 20.700 | 0.087 | 13.309 | 0.214 |
| RetinexNet + MIRNet* | 0.741 | 17.634 | 0.117 | 21.389 | 0.317 | LLFlow + MIRNet* | 0.821 | 20.887 | 0.086 | 13.181 | 0.194 |
| RetinexNet + $LPDM_{15}$ | 0.648 | 17.811 | 0.115 | 19.747 | 0.268 | LLFlow + $LPDM_{15}$ | 0.843 | **21.142** | **0.084** | **12.966** | 0.122 |
| RetinexNet + $LPDM_{30}$ | **0.760** | **18.034** | **0.112** | **19.590** | **0.198** | LLFlow + $LPDM_{30}$ | 0.828 | 21.038 | **0.084** | 12.976 | 0.137 |
| EnlightenGAN [17] | 0.652 | 17.483 | 0.135 | 20.009 | 0.322 | URetinex-Net [48] | 0.824 | 19.842 | 0.099 | 15.298 | **0.128** |
| EnlightenGAN + $BM3D_5$ | 0.715 | 17.332 | 0.139 | 20.558 | 0.220 | URetinex-Net + $BM3D_5$ | 0.824 | 19.784 | 0.099 | 15.281 | 0.156 |
| EnlightenGAN + $BM3D_{15}$ | 0.721 | 17.358 | 0.139 | 20.629 | 0.247 | URetinex-Net + $BM3D_{15}$ | 0.806 | 19.727 | 0.099 | 15.296 | 0.206 |
| EnlightenGAN + NAFNet | 0.741 | 17.618 | 0.133 | 20.083 | 0.242 | URetinex-Net + NAFNet | 0.791 | 19.487 | 0.101 | 15.489 | 0.216 |
| EnlightenGAN + MIRNet* | 0.760 | **17.774** | **0.132** | 20.026 | 0.232 | URetinex-Net + MIRNet* | 0.803 | 19.667 | 0.100 | 15.340 | 0.197 |
| EnlightenGAN + $LPDM_{15}$ | **0.761** | 17.574 | 0.136 | **19.975** | **0.173** | URetinex-Net + $LPDM_{15}$ | **0.827** | 20.017 | **0.097** | 14.859 | 0.137 |
| EnlightenGAN + $LPDM_{30}$ | 0.759 | 17.505 | 0.137 | 19.994 | 0.180 | URetinex-Net + $LPDM_{30}$ | 0.819 | **20.043** | **0.097** | **14.734** | 0.148 |
| KinD [45] | 0.771 | 17.648 | **0.123** | 18.031 | 0.175 | LLFormer [29] | 0.819 | 23.346 | 0.067 | 11.221 | 0.168 |
| KinD + $BM3D_5$ | 0.771 | 17.497 | 0.125 | 18.292 | 0.193 | LLFormer + $BM3D_5$ | 0.832 | 23.110 | 0.068 | 11.451 | 0.178 |
| KinD + $BM3D_{15}$ | 0.761 | 17.482 | 0.126 | 18.310 | 0.233 | LLFormer + $BM3D_{15}$ | 0.816 | 22.962 | 0.069 | 11.492 | 0.228 |
| KinD + NAFNet | 0.761 | 17.570 | 0.124 | 18.088 | 0.252 | LLFormer + NAFNet | 0.807 | 22.799 | 0.069 | 11.387 | 0.234 |
| KinD + MIRNet* | 0.770 | 17.605 | 0.124 | 18.066 | 0.240 | LLFormer + MIRNet* | 0.818 | 22.956 | 0.068 | 11.298 | 0.234 |
| KinD + $LPDM_{15}$ | **0.792** | **17.676** | 0.124 | 17.744 | **0.151** | LLFormer + $LPDM_{15}$ | **0.855** | **23.661** | **0.066** | **10.965** | **0.116** |
| KinD + $LPDM_{30}$ | 0.785 | 17.629 | 0.124 | **17.696** | 0.160 | LLFormer + $LPDM_{30}$ | 0.849 | 23.477 | 0.067 | 10.973 | 0.127 |
| KinD++ [8] | 0.758 | 17.752 | 0.113 | 17.773 | 0.198 | GSAD [30] | **0.851** | 22.728 | **0.075** | 12.191 | 0.134 |
| KinD++ + $BM3D_5$ | 0.768 | 17.714 | 0.113 | 17.799 | 0.203 | GSAD + $BM3D_5$ | 0.847 | 22.427 | 0.077 | 12.639 | 0.156 |
| KinD++ + $BM3D_{15}$ | 0.769 | 17.740 | 0.113 | 17.773 | 0.224 | GSAD + $BM3D_{15}$ | 0.833 | 22.329 | 0.077 | 12.666 | 0.194 |
| KinD++ + NAFNet | 0.765 | 17.768 | 0.114 | 17.647 | 0.245 | GSAD + NAFNet | 0.812 | 22.148 | 0.078 | 12.278 | 0.212 |
| KinD++ + MIRNet* | 0.776 | 17.816 | 0.113 | 17.780 | 0.237 | GSAD + MIRNet* | 0.828 | 22.418 | 0.077 | 12.318 | 0.192 |
| KinD++ + $LPDM_{15}$ | **0.794** | **17.867** | **0.112** | **17.528** | **0.165** | GSAD + $LPDM_{15}$ | **0.851** | **22.755** | **0.075** | **12.113** | **0.120** |
| KinD++ + $LPDM_{30}$ | 0.792 | 17.846 | **0.112** | 17.550 | 0.172 | GSAD + $LPDM_{30}$ | 0.838 | 22.615 | 0.076 | 12.154 | 0.128 |

the AdamW optimizer [59] with a learning rate of $1 \times 10^{-6}$ and with the AdamW parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\lambda = 0.01$. The loss function is defined in Eq. (5). We use the RGB color space for both low-light and normally-exposed images, and images are converted into the range $[-1, 1]$. We train on $256 \times 256$ random crops with random horizontal flipping. A batch size of 4 is used with an accumulation of gradients for 8 batches in order to simulate a batch size of 32.

The U-Net of LPDM consists of 4 downsampling stages (encoder) and 4 upsampling stages (decoder), with 2 residual blocks per stage. Between the encoder and the decoder is a middle block which processes the latent encoding. The middle block contains 2 residual blocks which surround a scaled dot-product attention layer using 8 attention heads. We avoid using attention mechanisms at higher resolutions than the final latent encoding in order to conserve memory. A visualization presenting more details of the LPDM architecture can be found in the supplementary materials. The implementation of the LPDM method is available online at https://github.com/savvaki/LPDM.

### 6.3. Benchmark study for LPDM

The following state-of-the-art LLIE approaches are selected for comparison: LIME [1], BIMEF [44], RetinexNet [24], EnlightenGAN [17], KinD [45], KinD++ [8], ZeroDCE [46], ZeroDCE++ [47], URetinex-Net [48], LLFlow [18], LLFormer [29] and GSAD [30]. BIMEF and LIME are non-learning-based methods and the remaining methods are all learning-based. GSAD is a state-of-the-art diffusion-based approach for LLIE [30]. The GSAD, LLFlow, LLFormer, URetinex-Net and RetinexNet methods are trained on the LOL dataset only. The KinD and KinD++ models are trained on LOL with additional custom synthetic data added [8]. The ZeroDCE and ZeroDCE++ models are trained on multi-exposure image sets from the SICE [60] dataset. EnlightenGAN is trained with unpaired groups of low-light and normal-exposure images using data from LOL as well as additional datasets.

Several methods such as KinD, KinD++, URetinex-Net, LLFlow and GSAD scale their model outputs based on an illumination ratio which involves the ground truth label. In order to achieve a fair comparison, we do not use the ground truths to scale these model outputs. Instead, we treat the LOL test set as unpaired data as would be the case in real-life scenarios. Similar to previous approaches, we fix the parameters of the LIME algorithm to $\alpha = 0.15$ and $\sigma = 2$ and $\gamma = 0.8$.

We compare the abovementioned LLIE approaches with and without our proposed LPDM. For the remainder of this work, $LPDM_s$ represents the LPDM approach applied with parameter $s$, defined in Section 5.3. All LPDM results fix $\phi = 300$, and we report two values of $s$ which are 15 and 30. Table 1 contains results with and without LPDM on the LOL test set represented as $\eta + LPDM_s$. The LPDM results are averaged over 5 runs with different seeds. In addition to the LOL dataset evaluation, we provide qualitative and quantitative results on the abovementioned unpaired test sets. The unpaired test metrics can be found in Table 2.

**Table 2**
Results on unpaired test sets for different LLIE methods ($\eta$), with and without the proposed LPDM. The results for LPDM are averaged over 5 runs.

| Method | LIME [1] | | | DICM [49] | | | MEF [50] | | | NPE [51] | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NIQE ↓ | BRISQUE ↓ | SPAQ ↑ | NIQE ↓ | BRISQUE ↓ | SPAQ ↑ | NIQE ↓ | BRISQUE ↓ | SPAQ ↑ | NIQE ↓ | BRISQUE ↓ | SPAQ ↑ |
| LIME [1] | **4.083** | 18.977 | 64.485 | 3.812 | 23.356 | 63.301 | **3.558** | **16.337** | 70.911 | **4.196** | 17.296 | 73.505 |
| LIME + $LPDM_{15}$ | 4.263 | **17.247** | 68.246 | **3.722** | 22.378 | 66.071 | 3.624 | 17.853 | **71.718** | 4.196 | **15.971** | 74.256 |
| LIME + $LPDM_{30}$ | 4.433 | 21.186 | **69.891** | 3.910 | **22.255** | **67.064** | 4.156 | 23.504 | 71.455 | 4.474 | 19.348 | **74.467** |
| BIMEF [44] | **3.859** | 17.714 | 65.121 | 3.827 | 23.588 | 64.375 | **3.330** | **15.262** | 67.352 | **4.135** | **14.928** | 72.622 |
| BIMEF + $LPDM_{15}$ | 4.199 | **17.452** | 68.349 | **3.812** | **20.587** | 68.164 | 3.731 | 18.245 | **68.789** | 4.164 | 15.461 | 74.402 |
| BIMEF + $LPDM_{30}$ | 4.330 | 20.875 | **69.098** | 4.004 | 21.329 | **69.120** | 4.345 | 21.830 | 68.677 | 4.421 | 17.775 | **75.008** |
| RetinexNet [24] | 4.598 | 26.341 | **70.370** | 4.451 | 29.339 | 68.805 | 4.416 | 20.068 | **74.386** | 4.595 | 22.579 | 74.910 |
| RetinexNet + $LPDM_{15}$ | **4.520** | **22.443** | 69.949 | 4.141 | 22.946 | 69.712 | **3.689** | **14.995** | 73.579 | **4.387** | **19.330** | 74.631 |
| RetinexNet + $LPDM_{30}$ | 4.594 | 24.306 | 70.031 | **4.046** | **22.903** | **70.167** | 4.112 | 23.055 | 72.502 | 4.573 | 22.422 | **74.958** |
| EnlightenGAN [17] | **3.657** | **14.879** | 64.208 | **3.561** | **18.808** | 63.124 | **3.221** | **14.332** | 68.701 | **4.116** | **15.344** | 73.552 |
| EnlightenGAN + $LPDM_{15}$ | 4.129 | 17.069 | 68.422 | 3.756 | 19.137 | 66.464 | 3.663 | 18.282 | 71.214 | 4.143 | 16.630 | 74.816 |
| EnlightenGAN + $LPDM_{30}$ | 4.226 | 20.419 | **69.574** | 3.975 | 22.096 | **67.274** | 4.234 | 22.501 | **71.280** | 4.366 | 18.004 | **74.999** |
| KinD [45] | 4.762 | 25.208 | 64.511 | 4.139 | 29.116 | 65.324 | **3.875** | 27.500 | 66.879 | **4.167** | 18.092 | 74.067 |
| KinD + $LPDM_{15}$ | **4.172** | 19.864 | 68.099 | 3.988 | 24.931 | 68.395 | 4.031 | 23.964 | 69.678 | 4.185 | **16.688** | **74.088** |
| KinD + $LPDM_{30}$ | 4.223 | **19.282** | **68.831** | **3.971** | **24.288** | **68.821** | 4.136 | **23.235** | **69.833** | 4.326 | 17.557 | 74.059 |
| KinD++ [8] | 4.726 | 23.298 | 65.992 | **3.786** | 26.314 | 68.086 | **3.737** | 28.635 | **70.250** | 4.384 | 20.970 | **75.817** |
| KinD++ + $LPDM_{15}$ | **4.221** | 21.624 | 67.907 | 3.835 | 24.673 | **68.936** | 3.870 | 26.107 | 70.190 | **4.270** | **18.694** | 74.966 |
| KinD++ + $LPDM_{30}$ | 4.300 | **21.062** | **68.705** | 3.868 | **24.516** | 68.913 | 4.076 | **25.285** | 69.890 | 4.369 | 20.192 | 74.495 |
| ZeroDCE [46] | **3.773** | 18.480 | 64.883 | **3.723** | 23.519 | 64.769 | **3.283** | **16.632** | **70.906** | **3.946** | **15.739** | 74.399 |
| ZeroDCE + $LPDM_{15}$ | 4.167 | **18.009** | 68.342 | 3.733 | **19.091** | 68.175 | 3.596 | 17.879 | 70.904 | 4.004 | 15.987 | 74.645 |
| ZeroDCE + $LPDM_{30}$ | 4.321 | 22.128 | **69.375** | 3.988 | 23.075 | **68.884** | 4.295 | 23.243 | 70.365 | 4.288 | 18.245 | **74.751** |
| ZeroDCE++ [47] | **3.965** | **17.217** | 64.956 | 3.824 | 21.368 | 64.470 | **3.400** | **13.602** | 70.291 | **4.021** | **12.941** | 73.658 |
| ZeroDCE++ + $LPDM_{15}$ | 4.240 | 17.585 | 68.727 | **3.799** | **18.738** | 68.024 | 3.686 | 17.642 | **70.663** | 4.106 | 15.305 | 74.685 |
| ZeroDCE++ + $LPDM_{30}$ | 4.347 | 21.639 | **69.725** | 3.983 | 22.877 | **68.790** | 4.291 | 22.986 | 70.240 | 4.351 | 17.971 | **74.825** |
| LLFlow [18] | 5.071 | 26.596 | 68.282 | **3.831** | 24.895 | 67.698 | **3.924** | 26.973 | **72.368** | **4.203** | 19.680 | 73.812 |
| LLFlow + $LPDM_{15}$ | 4.485 | 21.517 | 69.766 | 3.875 | 21.675 | 68.947 | 3.935 | 23.632 | 70.841 | 4.273 | 17.295 | 74.052 |
| LLFlow + $LPDM_{30}$ | **4.345** | **19.364** | **70.372** | 3.921 | **20.001** | **69.308** | 4.076 | **21.803** | 70.232 | 4.368 | **16.105** | **74.168** |
| URetinex-Net [48] | 4.353 | 23.906 | 68.232 | 4.152 | 24.261 | 66.239 | **3.790** | 21.478 | **72.005** | 4.688 | 24.384 | 74.462 |
| URetinex-Net + $LPDM_{15}$ | 4.393 | 22.362 | **69.126** | **3.932** | **23.294** | **66.965** | 3.942 | 22.381 | 70.986 | 4.498 | 22.040 | **74.472** |
| URetinex-Net + $LPDM_{30}$ | **4.316** | **20.732** | 69.086 | 3.951 | 23.323 | 66.887 | 4.041 | **21.079** | 70.219 | **4.484** | **22.013** | 74.349 |
| LLFormer [29] | 4.145 | **14.664** | 61.821 | 3.850 | **14.610** | 63.080 | **3.682** | **17.043** | 63.139 | **3.962** | **12.440** | 71.975 |
| LLFormer + $LPDM_{15}$ | **4.114** | 17.601 | 66.030 | **3.844** | 18.704 | 68.043 | 3.880 | 19.968 | 69.661 | 4.071 | 13.974 | 74.829 |
| LLFormer + $LPDM_{30}$ | 4.128 | 18.226 | **67.131** | 3.871 | 17.316 | **68.935** | 4.014 | 18.554 | **70.471** | 4.214 | 12.997 | **75.138** |
| GSAD [30] | 4.411 | 24.817 | 62.190 | 4.290 | **23.215** | **67.959** | 4.434 | 27.567 | 67.750 | 4.829 | 23.543 | 72.206 |
| GSAD + $LPDM_{15}$ | 4.216 | 22.000 | 65.801 | 4.104 | 25.511 | 67.006 | 4.244 | 27.550 | **68.536** | **4.296** | 21.203 | 73.487 |
| GSAD + $LPDM_{30}$ | **4.144** | **20.087** | **66.629** | **4.029** | 23.454 | 66.479 | **4.199** | **25.337** | 68.403 | 4.346 | **18.690** | **73.556** |

### 6.4. Comparison with alternative denoisers

As mentioned in Section 3, we compare the LPDM denoising performance with the popular BM3D [11] algorithm as well as the state-of-the-art DL denoisers NAFNet [12] and MIRNet* [13]. For the BM3D algorithm, we follow the denoising strategy used by LIME [1]. For all figures and tables, $BM3D_\sigma$ represents the application of the BM3D algorithm where $\sigma$ is the standard deviation parameter of BM3D. The results are reported in Table 1 as $\eta + BM3D_\sigma$.

NAFNet and MIRNet* are each evaluated as a post-processing step after LLIE. Both NAFNet and MIRNet* are used to denoise enhanced images $\hat{\boldsymbol{x}}_0^\eta$, and the outputs are captured for each $\eta$ over each test image. The results of NAFNet denoising can be found in Table 1 represented as $\eta$+NAFNet. Similarly, results for MIRNet* are reported as $\eta$+MIRNet*. Finally, we measure the latency of LPDM, NAFNet and BM3D on a compute node with an NVIDIA V100 GPU and 10 CPU cores. Random input is use at resolution 600 × 400 which is the size of a LOL test image. We were unable to calculate the latency of MIRNet* since the model uses more than 32 GB of GPU memory during inference.

### 6.5. Interpretation of results

The full-reference and no-reference metrics are reported in Tables 1 and 2, respectively. Metrics marked in bold indicate that they are the best for a particular method (ties are also marked in bold). The LOL test set results in Table 1 show that our LPDM is able to improve the SSIM of all baseline LLIE methods except LLFlow and GSAD on the LOL dataset. The GSAD SSIM remains the same, however the perceptual quality improved, which demonstrates that even existing diffusion approaches can be improved by LPDM. In all cases, LPDM improves the PSNR compared to each baseline. In many cases, the SSIM is greatly improved by adding LPDM. Adding LPDM to LIME, RetinexNet, EnlightenGAN, ZeroDCE, ZeroDCE++ and LLFormer boasts up to a 53%, 78%, 16%, 23%, 23%, 4% SSIM improvement, respectively. LLFormer yields new state-of-the-art SSIM results on the LOL dataset when LPDM post-processing is added.

Importantly, LPDM outperforms the alternative post-processing denoisers on the perceptual LPIPS metric in all cases. We emphasize LPIPS since it correlates more with human perceptual quality than the other quantitative metrics chosen [54]. This is further confirmed by the color error $\Delta E_s$, which shows that LPDM is able to improve the accuracy of color information of most DL methods. LPDM is able to improve the baseline PSNR for all methods, and mostly outperforms the competing denoisers on the PSNR metric. In some cases, NAFNet and MIRNet* are able to improve PSNR and MAE more than our LPDM; however, upon further inspection, this is as a result of aggressive denoising and thus oversmoothing. PSNR and MAE favor the smooth solutions predicted by the alternative denoisers since these metrics do not consider image structure or perceptual quality [61]. Fig. 6 displays a comparison of post-processing approaches. NAFNet and MIRNet* remove most typical noises, however at the cost of removing detail. For example, consider the first row of Fig. 6, where technically NAFNet and MIRNet* yield a higher PSNR for $\hat{\boldsymbol{x}}_0^{\text{LIME}}$, but the results are clearly blurred when compared to LPDM which maintains the sharpness of the original image. The behavior of NAFNet and MIRNet* manifests more accurately in the LPIPS metric where the LPDM outperforms these models significantly


**Fig. 6.** Qualitative comparison of the BM3D [11], NAFNet [12] and MIRNet* [13] post-processing denoising approaches to LPDM on the LOL test set. The first row displays examples of enhanced low-light images ($\hat{\boldsymbol{x}}_0^{\eta}$) for different LLIE approaches, and the remaining rows display the results of post-processing denoisers applied to each $\hat{\boldsymbol{x}}_0^{\eta}$.

for all LLIE methods. Although the BM3D denoiser performs well, it is unable to deal with color noises and other distortions as robustly as DL methods.

The alternative post-processing approaches are unable to generalize beyond the types of noise on which they are trained. In contrast, our method models the conditional distribution between low-light and normally-exposed images, allowing LPDM to handle a variety of different artifacts and color distortions besides typical noise. An example of a distortion which differs from typical Gaussian noise is the distortion introduced by KinD++. Fig. 6 shows that our LPDM increases the sharpness of $\hat{\boldsymbol{x}}_0^{\text{KinD++}}$ where the other denoisers yield oversmoothed results. We emphasize this point because different LLIE methods introduce a panoply of different distortions. Finally, note that post-processing approaches generally run the risk of further degrading an image, especially if the input image is already well-enhanced. However, the metrics in Table 1 suggest that LPDM is less susceptible to negatively impacting the input image when compared with the alternative denoisers.

For the majority of methods and datasets in Table 2, LPDM is able to improve the SPAQ score. The improvement of the NIQE score and BRISQUE score fluctuates depending on the dataset and the method. However, NIQE and BRISQUE are fickle metrics since their generalization ability is limited by the scope of the natural data on which these models are trained. SPAQ is a more reliable metric since it employs a pretrained DL transformer, and has demonstrated state-of-the-art image quality analysis performance [57]. Nonetheless, it still remains vital to analyze the qualitative effects of LPDM on real test data. Several

**Table 3**
Ablation study comparing LPDM to DLPDM and ULPDM on the LOL test set. The results for each model are averaged over 5 runs.

| Methods | SSIM (%) ↑ | PSNR (%) ↑ | MAE (%) ↑ | $\Delta E_s$ (%) ↑ | LPIPS (%) ↑ |
|---|---|---|---|---|---|
| DLPDM | $16.956 \pm 26.46$ | $1.9 \pm 3.72$ | $1.827 \pm 5.17$ | $2.8 \pm 5.18$ | $2.237 \pm 31.14$ |
| $\text{ULPDM}_{15}$ | $3.764 \pm 26.2$ | $-0.335 \pm 2.3$ | $-0.648 \pm 3.06$ | $-1.338 \pm 1.27$ | $-4.621 \pm 33.98$ |
| $\text{ULPDM}_{30}$ | $-13.224 \pm 26.9$ | $-1.935 \pm 3.03$ | $-2.476 \pm 4.3$ | $-2.47 \pm 2.43$ | $-35.305 \pm 45.28$ |
| $\text{LPDM}_{15}$ | $15.498 \pm 17.46$ | $1.139 \pm 1.74$ | $0.879 \pm 2.79$ | $0.945 \pm 1.4$ | $27.248 \pm 20.03$ |
| $\text{LPDM}_{30}$ | $17.964 \pm 25.0$ | $0.934 \pm 2.23$ | $0.554 \pm 3.52$ | $1.004 \pm 1.72$ | $25.099 \pm 25.74$ |

images from a variety of datasets are displayed in Fig. 1. The general advantage of LPDM is its ability to strike a balance between smoothing and maintaining sharpness. Due to the distribution of the noise output of LPDM being zero-centered, our approach maintains the perceptual quality of the underlying image and avoids oversmoothing when Eq. (8) is applied. In many cases, LPDM is able to improve color quality and sharpness. Upon close inspection, LPDM alters color shades to more accurately represent reality, as seen for BIMEF, and LLFormer examples in Fig. 1. More examples are included in the supplementary materials due to space limitations.

In addition to the above conclusions, the choices of $s = 15$ and $s = 30$ may not necessarily be the optimal values for each LLIE method. Therefore, there may be larger improvements for a different choice of $s$ which can be determined empirically. We fix $s$ in order to demonstrate the possibility of using LPDM as a blind denoiser.

Finally, we measure the latency of LPDM at 228 ms (ms) for inputs of size $600 \times 400$. The latency of NAFNet and BM3D is measured at 98 ms and 3132 ms, respectively. Although LPDM is somewhat slower than NAFNet, the speed of LPDM is still viable, especially when denoising performance is considered. In contrast, traditional iterative DM approaches would be infeasible due to their iterative nature. Future optimization and compression of the LPDM framework may incur significant speedups. BM3D is the slower algorithm of those compared; however, more optimized alternative versions of the BM3D algorithm may improve computational performance.

### 6.6. Ablation study

An ablation study is necessary in order to demonstrate that the improvements of LPDM can be attributed specifically to our proposed approach and that the results are not arbitrary. In Section 6.6.1 we examine the effect of predicting $\epsilon$, and in Section 6.6.2 we compare unconditional diffusion to LPDM. In order to conserve space, tables for the ablation study results are summarized such that the percentage improvement for each metric is calculated for each $\eta$, and the mean and standard deviation percentage improvements are reported.

#### 6.6.1. The effect of predicting the noise

As seen in Eq. (5), DMs are trained to make predictions for $\epsilon$. We examine the value of predicting $\epsilon$ by changing the model to predict $\boldsymbol{x}_0$ directly, and we name this model *direct* LPDM or DLPDM. The DLPDM model directly denoises the input and does not require any further steps such as Eq. (8). The DLPDM is identical to LPDM described in Section 6.2 with two differences: the ground truth target of the model is now $\boldsymbol{x}_0$ rather than $\epsilon$, and we remove timestep conditioning from the model by setting $t = 0$ as input to the model regardless of $\boldsymbol{x}_t$. Therefore, the model directly denoises its input with the same number of parameters and without the requirement of specifying $\phi$ at inference time, thus making the model a blind denoiser. In other terms, the model is responsible for detecting the amount of noise present in $\boldsymbol{x}_t$ and predicting $\boldsymbol{x}_0$ without any additional parameters defined by the user.

The results of the DLPDM experiment are summarized in Table 3, which includes the other ablation results from Section 6.6.2. Each model is averaged over 5 runs with different seeds. Our proposed LPDM approach performs better on SSIM and LPIPS and DLPDM performs better on PSNR and MAE (although the variance of DLPDM is higher). LPDM largely outperforms DLPDM on LPIPS which implies that LPDM results are more perceptually similar to the ground truth. Both DLPDM and LPDM improve $\Delta E_s$ on average, however the standard deviation of DLPDM is higher, making it less reliable for restoring color details. The results are verified when examining the examples in Fig. 7 where LPDM preserves the sharpness of $\hat{\boldsymbol{x}}_0^{\eta}$ and maintains color accuracy.

#### 6.6.2. The effect of conditioning

We explore the effect of appending $\boldsymbol{c}$ to $\boldsymbol{x}_t$ as visually depicted in Fig. 3. We name this model *unconditional* LPDM or ULPDM. In many cases, diffusion models may ignore the concatenated conditioning and simply learn how to denoise. Therefore, it is important to explore whether LPDM requires the use of conditioning to achieve the desired results. ULPDM is an identical model to LPDM from Section 6.2, however, we change the input layer to accept only $\boldsymbol{x}_t$ as input, thus changing the number of input channels from six to three. In other terms, we compare conditional diffusion to unconditional diffusion.

The experimental results in Table 3 show that LPDM significantly outperforms ULPDM across all metrics. Therefore, we conclude that conditioning is necessary in order for LPDM to detect the wide variety of artifacts that can be present in $\hat{\boldsymbol{x}}_0^{\eta}$. We provide visual results in Fig. 8 which verify our conclusion: ULPDM is able to remove noise, however results are oversmoothed and thus detail is lost due to lack of conditioning.

## 7. Conclusion

In this paper, we present a framework for post-processing images which have undergone low-light image enhancement. The enhancement of low-light images often reveals a variety of degradations which are hidden in the dark, and thus a need for post-processing is introduced. Furthermore, each low-light enhancement technique can possibly introduce a different form of degradation into its result. We propose using a conditional diffusion model in order to model the distribution between under-exposed and normally-exposed images. Further, we introduce a method of applying our diffusion model as a non-iterative post-processing technique, which is unlike existing diffusion approaches. Our approach uses the diffusion model to estimate the amount of noise present in an enhanced image in one pass through the model, which can simply be subtracted from the enhanced image to further enhance the image. The subtraction of the noise is parameterized via $s$ and $\phi$, and experiments show that $\phi$ may be fixed. The parameter $s$ may be selected by the user, or fixed to a small value for convenience as we have empirically demonstrated. Our approach is unlike conventional diffusion generative sampling since it repurposes the diffusion model for efficient non-iterative noise detection.

We compare our method to 3 alternative denoisers and apply our post-processing approach to 12 state-of-the-art low-light image enhancement backbones. We show that our approach can improve the output of state-of-the-art low-light image enhancers in terms of perceptual quality. In contrast to existing denoisers, our approach is less susceptible to oversmoothing and is capable of enhancing perceptual quality while eliminating noise and other distortions. Our technique can be used to repair enhanced low-light images given any arbitrary low-light image enhancer backbone. In future work, we will investigate whether the hyperparameters introduced by our approach may be eliminated and determine if the model size can be reduced. The applicability of our approach to other image restoration domains is a promising avenue for future research.

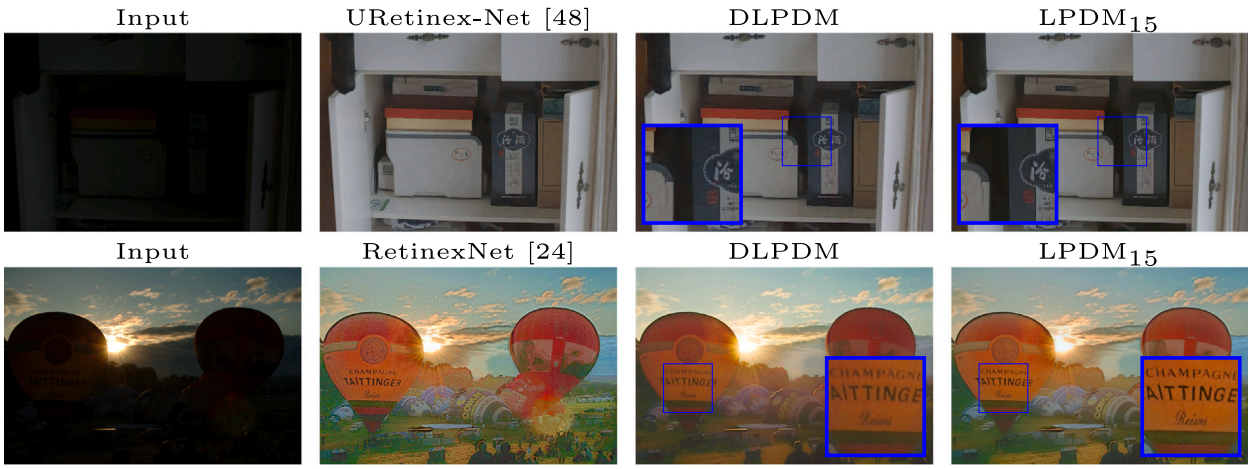


**Fig. 7.** Visual examples of the ablation study directly comparing predicting $x_0$ to predicting $\epsilon$ using the DLPDM and LPDM models, respectively.

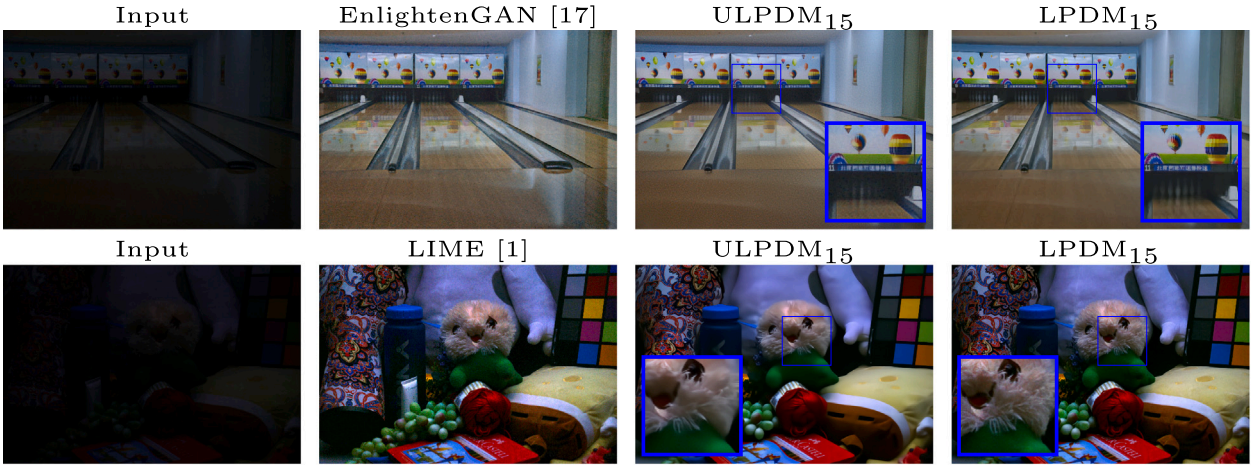


**Fig. 8.** Visual examples of the ablation study comparing unconditional and conditional diffusion using the ULPDM and LPDM models, respectively.

## CRediT authorship contribution statement

**Savvas Panagiotou:** Conceptualization, Data curation, Investigation, Methodology, Project administration, Resources, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Anna S. Bosman:** Funding acquisition, Supervision, Writing – review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

The public link to the code is specified in the manuscript. The datasets are listed in the manuscript and are open source/widely available.

## Acknowledgments

The financial assistance of the National Research Foundation (NRF) of South Africa towards this research is hereby acknowledged. The research has been supported by the NRF Thuthuka grant number TTK2103165901115. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF. The authors acknowledge the Centre for High Performance Computing (CHPC), South Africa, for providing computational resources to this research project.

## Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.patcog.2024.110799.

## References

[1] X. Guo, Y. Li, H. Ling, LIME: Low-light image enhancement via illumination map estimation, IEEE Trans. Image Process. 26 (2) (2017) 982–993, http://dx.doi.org/10.1109/TIP.2016.2639450.

[2] Y. Sasagawa, H. Nagahara, Yolo in the dark-domain adaptation method for merging multiple models, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16, Springer, 2020, pp. 345–359.

[3] J. Liu, D. Xu, W. Yang, M. Fan, H. Huang, Benchmarking low-light image enhancement and beyond, Int. J. Comput. Vis. 129 (2021) 1153–1184.

[4] K.G. Lore, A. Akintayo, S. Sarkar, LLNet: A deep autoencoder approach to natural low-light image enhancement, Pattern Recognit. 61 (2017) 650–662.

[5] S. Zheng, G. Gupta, Semantic-guided zero-shot learning for low-light image/video enhancement, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 581–590.

[6] S.M. Pizer, E.P. Amburn, J.D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J.B. Zimmerman, K. Zuiderveld, Adaptive histogram equalization and its variations, Comput. Vis. Graph. Image Process. 39 (3) (1987) 355–368.

[7] C. Chen, Q. Chen, J. Xu, V. Koltun, Learning to see in the dark, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3291–3300.

[8] Y. Zhang, X. Guo, J. Ma, W. Liu, J. Zhang, Beyond brightening low-light images, Int. J. Comput. Vis. 129 (2021) 1013–1037.

[9] Y. Zhou, J. Jiao, H. Huang, Y. Wang, J. Wang, H. Shi, T. Huang, When awgn-based denoiser meets real noises, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 13074–13081.

[10] L. Li, R. Wang, W. Wang, W. Gao, A low-light image enhancement method for both denoising and contrast enlarging, in: 2015 IEEE International Conference on Image Processing, ICIP, IEEE, 2015, pp. 3730–3734.

[11] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3-D transform-domain collaborative filtering, IEEE Trans. Image Process. 16 (8) (2007) 2080–2095.

[12] L. Chen, X. Chu, X. Zhang, J. Sun, Simple baselines for image restoration, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII, Springer, 2022, pp. 17–33.

[13] Y. Cai, X. Hu, H. Wang, Y. Zhang, H. Pfister, D. Wei, Learning to generate realistic noisy images via pixel-level noise-aware adversarial training, Adv. Neural Inf. Process. Syst. 34 (2021) 3259–3270.

[14] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, in: International Conference on Machine Learning, PMLR, 2015, pp. 2256–2265.

[15] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 6840–6851, URL https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

[16] P. Dhariwal, A. Nichol, Diffusion models beat GANs on image synthesis, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J.W. Vaughan (Eds.), Advances in Neural Information Processing Systems, Vol. 34, Curran Associates, Inc., 2021, pp. 8780–8794, URL https://proceedings.neurips.cc/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf.

[17] Y. Jiang, X. Gong, D. Liu, Y. Cheng, C. Fang, X. Shen, J. Yang, P. Zhou, Z. Wang, Enlightengan: Deep light enhancement without paired supervision, IEEE Trans. Image Process. 30 (2021) 2340–2349.

[18] Y. Wang, R. Wan, W. Yang, H. Li, L.-P. Chau, A. Kot, Low-light image enhancement with normalizing flow, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, pp. 2604–2612.

[19] J.A. Stark, Adaptive image contrast enhancement using generalizations of histogram equalization, IEEE Trans. Image Process. 9 (5) (2000) 889–896.

[20] S. Rahman, M.M. Rahman, M. Abdullah-Al-Wadud, G.D. Al-Quaderi, M. Shoyaib, An adaptive gamma correction for image enhancement, EURASIP J. Image Video Process. 2016 (1) (2016) 1–13.

[21] E.H. Land, J.J. McCann, Lightness and retinex theory, JOSA 61 (1) (1971) 1–11.

[22] M. Li, J. Liu, W. Yang, X. Sun, Z. Guo, Structure-revealing low-light image enhancement via robust retinex model, IEEE Trans. Image Process. 27 (6) (2018) 2828–2841.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).

[24] C. Wei, W. Wang, W. Yang, J. Liu, Deep retinex decomposition for low-light enhancement, in: British Machine Vision Conference, 2018.

[25] Z. Zhang, H. Zheng, R. Hong, M. Xu, S. Yan, M. Wang, Deep color consistent network for low-light image enhancement, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 1899–1908.

[26] Z. Ji, H. Zheng, Z. Zhang, Q. Ye, Y. Zhao, M. Xu, Multi-scale interaction network for low-light stereo image enhancement, IEEE Trans. Consum. Electron. (2023).

[27] S. Zheng, Y. Ma, J. Pan, C. Lu, G. Gupta, Low-light image and video enhancement: A comprehensive survey and beyond, 2022, arXiv preprint arXiv:2212.10772.

[28] S.W. Zamir, A. Arora, S. Khan, M. Hayat, F.S. Khan, M.-H. Yang, Restormer: Efficient transformer for high-resolution image restoration, in: CVPR, 2022.

[29] T. Wang, K. Zhang, T. Shen, W. Luo, B. Stenger, T. Lu, Ultra-high-definition low-light image enhancement: A benchmark and transformer-based method, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 2654–2662.

[30] J. Hou, Z. Zhu, J. Hou, H. Liu, H. Zeng, H. Yuan, Global structure-aware diffusion process for low-light image enhancement, in: Advances in Neural Information Processing Systems, 36, 2024.

[31] I. Kobyzev, S.J. Prince, M.A. Brubaker, Normalizing flows: An introduction and review of current methods, IEEE Trans. Pattern Anal. Mach. Intell. 43 (11) (2020) 3964–3979.

[32] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, Springer, 2015, pp. 234–241.

[33] A. Abdelhamed, S. Lin, M.S. Brown, A high-quality denoising dataset for smartphone cameras, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1692–1700.

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

[35] A.Q. Nichol, P. Dhariwal, Improved denoising diffusion probabilistic models, in: M. Meila, T. Zhang (Eds.), Proceedings of the 38th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 139, PMLR, 2021, pp. 8162–8171, URL https://proceedings.mlr.press/v139/nichol21a.html.

[36] J. Ho, T. Salimans, Classifier-free diffusion guidance, in: NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications, 2021, URL https://openreview.net/forum?id=qw8AKxfYbI.

[37] J. Song, C. Meng, S. Ermon, Denoising diffusion implicit models, in: International Conference on Learning Representations, 2021, URL https://openreview.net/forum?id=St1giarCHLP.

[38] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10684–10695.

[39] B. Kim, Y. Oh, J.C. Ye, Diffusion adversarial representation learning for self-supervised vessel segmentation, in: International Conference on Learning Representations, 2023, URL https://openreview.net/forum?id=H0gdPxSwkPb.

[40] B. Kim, I. Han, J.C. Ye, DiffuseMorph: Unsupervised deformable image registration using diffusion model, in: S. Avidan, G. Brostow, M. Cissé, G.M. Farinella, T. Hassner (Eds.), Computer Vision – ECCV 2022, Springer Nature Switzerland, Cham, ISBN: 978-3-031-19821-2, 2022, pp. 347–364.

[41] T. Salimans, A. Karpathy, X. Chen, D.P. Kingma, PixelCNN++: Improving the pixelCNN with discretized logistic mixture likelihood and other modifications, in: Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, 2017, URL https://openreview.net/forum?id=BJrFC6ceg.

[42] Y. Wu, K. He, Group normalization, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 3–19.

[43] S. Elfwing, E. Uchibe, K. Doya, Sigmoid-weighted linear units for neural network function approximation in reinforcement learning, Neural Netw. 107 (2018) 3–11.

[44] Z. Ying, G. Li, W. Gao, A bio-inspired multi-exposure fusion framework for low-light image enhancement, 2017, arXiv preprint arXiv:1711.00591.

[45] Y. Zhang, J. Zhang, X. Guo, Kindling the darkness: A practical low-light image enhancer, in: Proceedings of the 27th ACM International Conference on Multimedia, 2019, pp. 1632–1640.

[46] C. Guo, C. Li, J. Guo, C.C. Loy, J. Hou, S. Kwong, R. Cong, Zero-reference deep curve estimation for low-light image enhancement, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020.

[47] C. Li, C. Guo, C.C. Loy, Learning to enhance low-light image via zero-reference deep curve estimation, IEEE Trans. Pattern Anal. Mach. Intell. 44 (8) (2021) 4225–4238.

[48] W. Wu, J. Weng, P. Zhang, X. Wang, W. Yang, J. Jiang, URetinex-Net: Retinex-based deep unfolding network for low-light image enhancement, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 5891–5900, http://dx.doi.org/10.1109/CVPR52688.2022.00581.

[49] C. Lee, C. Lee, C.-S. Kim, Contrast enhancement based on layered difference representation of 2D histograms, IEEE Trans. Image Process. 22 (12) (2013) 5372–5384, http://dx.doi.org/10.1109/TIP.2013.2284059.

[50] K. Ma, K. Zeng, Z. Wang, Perceptual quality assessment for multi-exposure image fusion, IEEE Trans. Image Process. 24 (11) (2015) 3345–3356, http://dx.doi.org/10.1109/TIP.2015.2442920.

[51] S. Wang, J. Zheng, H.-M. Hu, B. Li, Naturalness preserved enhancement algorithm for non-uniform illumination images, IEEE Trans. Image Process. 22 (9) (2013) 3538–3548, http://dx.doi.org/10.1109/TIP.2013.2261309.

[52] W. Yang, W. Wang, H. Huang, S. Wang, J. Liu, Sparse gradient regularized deep retinex network for robust low-light image enhancement, IEEE Trans. Image Process. 30 (2021) 2072–2086.

[53] X. Zhang, B.A. Wandell, A spatial extension of CIELAB for digital color-image reproduction, J. Soc. Inf. Disp. 5 (1) (1997) 61–63, http://dx.doi.org/10.1889/1.1985127.

[54] R. Zhang, P. Isola, A.A. Efros, E. Shechtman, O. Wang, The unreasonable effectiveness of deep features as a perceptual metric, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 586–595.

[55] A. Mittal, R. Soundararajan, A.C. Bovik, Making a “completely blind” image quality analyzer, IEEE Signal Process. Lett. 20 (3) (2013) 209–212.

[56] A. Mittal, A.K. Moorthy, A.C. Bovik, No-reference image quality assessment in the spatial domain, IEEE Trans. Image Process. 21 (12) (2012) 4695–4708.

[57] J. Ke, Q. Wang, Y. Wang, P. Milanfar, F. Yang, Musiq: Multi-scale image quality transformer, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 5148–5157.

[58] Y. Fang, H. Zhu, Y. Zeng, K. Ma, Z. Wang, Perceptual quality assessment of smartphone photography, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 3677–3686.
[59] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: International Conference on Learning Representations, 2019, URL https://openreview.net/forum?id=Bkg6RiCqY7.
[60] J. Cai, S. Gu, L. Zhang, Learning a deep single image contrast enhancer from multi-exposure images, IEEE Trans. Image Process. 27 (4) (2018) 2049–2062.
[61] Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612, http://dx.doi.org/10.1109/TIP.2003.819861.

**Savvas Panagiotou** received an M.Sc. Computer Science from the University of Pretoria, South Africa. His research interests include deep learning and computer vision.

**Anna Bosman** is currently appointed as a senior lecturer in the Department of Computer Science, University of Pretoria, South Africa, where she received the Ph.D. degree in Computer Science in 2019. Her research interests include artificial neural networks, deep learning, neural architecture search, loss landscape analysis, and computer vision.