BAYESIAN ENHANCEMENT MODELS FOR ONE-TO-MANY MAPPING IN IMAGE ENHANCEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Image enhancement is considered an ill-posed inverse problem due to its tendency to have multiple solutions. The loss of information makes accurately reconstructing the original image from observed data challenging. Also, the quality of the result is often subjective to individual preferences. This obviously poses a one-to-many mapping challenge. To address this, we propose a Bayesian Enhancement Model (BEM) that leverages Bayesian estimation to capture inherent uncertainty and accommodate diverse outputs. To address the noise in predictions of Bayesian Neural Networks (BNNs) for high-dimensional images, we propose a two-stage approach. The first stage utilises a BNN to model reduced-dimensional image representations, while the second stage employs a deterministic network to refine these representations. We further introduce a dynamic *Momentum Prior* to overcome convergence issues typically faced by BNNs in high-dimensional spaces. Extensive experiments across multiple low-light and underwater image enhancement benchmarks demonstrate the superiority of our method over traditional deterministic models, particularly in real-world applications lacking reference images, highlighting the potential of Bayesian models in handling one-to-many mapping problems.

1 INTRODUCTION

In computer vision, image enhancement refers to the process of enhancing the perceptual quality, visibility, and overall appearance of an image, which can involve reducing noise, increasing contrast, sharpening details, or correcting colour imbalances. In image enhancement tasks such as low-light image enhancement (LLIE) and underwater image enhancement (UIE), a common challenge arises from dynamic photography conditions, where a single degraded input image can correspond to multiple plausible target images. This phenomenon, known as the *one-to-many mapping* problem, arises because multiple valid outputs can be generated depending on varying conditions during image capture, such as changes in lighting, exposure, or other factors.

Recent advances in deep learning have shifted image enhancement towards data-driven approaches. Several deep learning-based models (Zamir et al., 2022; Cai et al., 2023) have achieved advanced results by learning mappings between low-quality (LQ) inputs and their high-quality (HQ) counterparts using paired datasets. However, we observe that existing datasets exhibit the one-to-many relationship between their input and target domains. Specifically, we observe cases where there exist at least two image pairs with input images that are either identical or visually indistinguishable, yet their corresponding targets exhibit notable variations. When such discrepancies arise due to ambiguity in the target domain, a traditional deep neural network—being a deterministic function—struggles to effectively model these one-to-many image pairs. Previous methods employing deterministic neural networks (DNNs) for image enhancement often overlook this class of one-to-many samples, leading to sub-optimal solutions. Figure 1 (middle) demonstrates how a deterministic neural network trained on one-to-many mapping data struggles to predict any specific target, instead producing an averaged output due to "regression toward the mean".

To tackle the inherent ambiguity in image enhancement tasks caused by one-to-many mappings, we adopt a Bayesian framework that models these mappings probabilistically. Rather than relying on a sub-optimal deterministic approach, our method leverages Bayesian inference to sample multiple sets of network weights from a learned distribution, effectively creating a diverse ensemble of deep networks. Each sampled network captures a distinct plausible solution, allowing our model to map a single input to a distribution of possible target outputs. This approach theoretically enables the

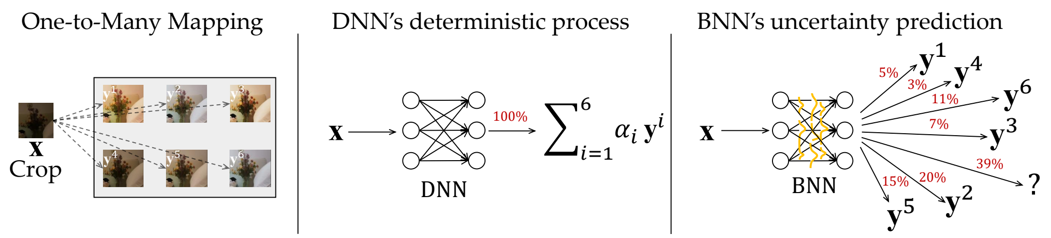


Figure 1: One-to-Many Mapping. The left panel shows an image crop $\mathbf{x}$ associated with multiple targets $\{\mathbf{y}^1, \ldots, \mathbf{y}^6\}$. A DNN (middle) trained on such data tends to predict the weighted average of all targets. In contrast, a BNN (right) models the one-to-many relation by producing different outputs according to a learned probability distribution.

mapping of all plausible variations, effectively modelling the complex one-to-many relationships present in real-world scenarios.

While BNNs have shown promise in capturing uncertainty in various tasks (Kendall & Cipolla, 2016; Kendall et al., 2015; 2018; Pang et al., 2020), their potential in addressing the one-to-many mapping problem for image enhancement remains largely under-explored. By incorporating Bayesian inference into the enhancement process, our approach captures uncertainty in dynamic, uncontrolled environments, providing a more flexible and robust solution than traditional deterministic models. However, applying BNNs to these tasks presents significant challenges due to the high dimensionality of image data and the strong 2D spatial correlations between pixels: The weight uncertainty in BNNs often leads to noisy image outputs, while models with high-dimensional weight spaces are prone to underfitting (Dusenberry et al., 2020; Tomczak et al., 2021). To mitigate the noise in BNN predictions, we propose a two-stage approach that combines a BNN and a DNN (Sec. 4). Following our approach, we systematically address these challenges, unleashing the potential of BNNs in low-light and underwater enhancement tasks.

As the first work to explore the feasibility of BNNs for image enhancement, we selected tasks where the *one-to-many mapping* problem is particularly pronounced, such as LLIE and UIE, to effectively validate our theoretical framework. The main contributions of this paper are summarised as follows:

- We identify the one-to-many mapping issue between inputs and outputs as a primary bottleneck in image enhancement models for LLIE and UIE, and propose the first Bayesian-based Enhancement Model (BEM) to learn this mapping relation.
- We introduce a dynamic prior called the *Momentum Prior* to mitigate the convergence difficulties typically encountered by BNNs in high-dimensional weight spaces.
- To reduce the complexity of BEM in modelling high-dimensional image data, we propose an innovative two-stage approach that combines the strengths of Bayesian NNs and Deterministic NNs).

2 Related Work

Bayesian Deep Learning. BNNs quantify uncertainty by learning distributions over network weights, offering robust predictions (Neal, 2012). Variational Inference (VI) is a common method for approximating these distributions (Graves, 2011; Blundell et al., 2015). Gal & Ghahramani (2016) simplify the implementation of BNNs by interpreting dropout as an approximate Bayesian inference method. Recent advancements show that adding uncertainty only to the final layer can efficiently approximate a full BNN (Harrison et al., 2024). Another line of approaches, such as Krishnan et al. (2020), explored the use of empirical Bayes to specify weight priors in BNNs to enhance the model's adaptability to diverse datasets. These BNN approaches have shown promise across a range of vision applications, including camera relocalisation (Kendall & Cipolla, 2016), semantic and instance segmentation (Kendall et al., 2015; 2018). Despite these advances, BNNs remain underutilised in image enhancement tasks.

Probabilistic Models in Image Enhancement. Several works have utilised probabilistic models to address different aspects of image enhancement. Jiang et al. (2021) employed GANs to capture features for LLIE, while Fabbri et al. (2018) leveraged CycleGAN (Zhu et al., 2017) to generate synthetic paired datasets, addressing data scarcity in UIE. FUnIE-GAN (Islam et al., 2020) further demonstrated effectiveness in both paired and unpaired UIE training. Anantrasirichai & Bull (2021) applied unpaired learning for LLIE when the scene conditions are known. Wang et al. (2022) applied normalising flow-based methods to reduce residual noise in LLIE predictions. However, its invertibility constraint limits model complexity. Zhou et al. (2024) mitigated this by integrating normalising flows with codebook techniques, introducing latent normalising flows. Diffusion Models (DMs) have been widely adopted for enhancement tasks (Hou et al., 2024; Tang et al., 2023). While DMs inherently address one-to-many mappings, their high latency for generating a single sample makes producing hundreds of candidates impractical due to prohibitive delays. Due to the practical limitations in generating multiple candidates, DM-based methods often prefer to produce an average of multiple targets, as this helps reduce the quality fluctuations within a single sampling process, as suggested by Jiang et al. (2023a).

2.1 Preliminaries

In image enhancement, the output of a neural network can be interpreted as the conditional probability distribution of the target image, $\mathbf{y} \in \mathcal{Y}$, given the degraded input image $\mathbf{x} \in \mathcal{X}$, and the network's weights $\mathbf{w}$: $P(\mathbf{y}|\mathbf{x}, \mathbf{w})$. Assuming the prediction errors follow a Gaussian distribution, the conditional probability density function (PDF) of the target image $\mathbf{y}$ can be modeled as a multivariate Gaussian, where the mean is given by the neural network output $F(\mathbf{x}; \mathbf{w})$:

$$P(\mathbf{y}|\mathbf{x}, \mathbf{w}) = \mathcal{N}(\mathbf{y}|F(\mathbf{x}; \mathbf{w}), \mathrm{diag}(\boldsymbol{\sigma}^2)). \tag{1}$$

The network weights $\mathbf{w}$ can be learned through maximum likelihood estimation (MLE). Given a dataset of image pairs $\{\mathbf{x}^i, \mathbf{y}^i\}_{i=1}^N$, the MLE estimate $\mathbf{w}^{\mathrm{MLE}}$ is computed by maximising the log-likelihood of the observed data:

$$\mathbf{w}^{\mathrm{MLE}} = \operatorname*{argmax}_{\mathbf{w}} \sum_{i=1}^{N} \log P(\mathbf{y}^i|\mathbf{x}^i, \mathbf{w}), \tag{2}$$

By optimising such an objective function in Eq. (2), the network $F_{\mathbf{w}}$ learns an injective function, $F_{\mathbf{w}} : \mathcal{X} \to \mathcal{Y}$. The deterministic nature of such a mapping implies that when $\mathbf{y}^i \neq \mathbf{y}^j$, the condition $\mathbf{x}^i \neq \mathbf{x}^j$ must hold. We argue that this deterministic process is inadequate in cases where one input corresponds to multiple plausible targets. In Sec. 3, we delve into methods for addressing this issue.

3 Modelling the one-to-many mapping

3.1 Bayesian Enhancement Models

We introduce uncertainty into the network weights $\mathbf{w}$ through Bayesian estimation, thus obtaining a posterior distribution over the weight, $\mathbf{w} \sim P(\mathbf{w}|\mathbf{y}, \mathbf{x})$. During inference, weights are sampled from this distribution. The posterior distribution over the weights is expressed as:

$$P(\mathbf{w}|\mathbf{y}, \mathbf{x}) = \frac{P(\mathbf{y}|\mathbf{x}, \mathbf{w})P(\mathbf{w})}{P(\mathbf{y}|\mathbf{x})} \tag{3}$$

where $P(\mathbf{y} \mid \mathbf{x}, \mathbf{w})$ represents the likelihood of observing $\mathbf{y}$ given the input $\mathbf{x}$ and weights $\mathbf{w}$, $P(\mathbf{w})$ denotes the prior distribution of the weights, and $P(\mathbf{y} \mid \mathbf{x})$ is the marginal likelihood.

Unfortunately, for any neural networks the posterior in Eq. (3) cannot be calculated analytically. This makes it impractical to directly sample weights from the true posterior distribution. Instead, we can leverage variational inference (VI) to approximate $P(\mathbf{w}|\mathbf{y}, \mathbf{x})$ with a more tractable distribution $q(\mathbf{w}|\boldsymbol{\theta})$. Such that, we can draw samples of weights $\mathbf{w}$ from the distribution $q(\mathbf{w}|\boldsymbol{\theta})$. As suggested by (Hinton & Van Camp, 1993; Graves, 2011; Blundell et al., 2015), the variational approximation is fitted by minimising their Kullback-Leibler (KL) divergence:

$$\begin{aligned}
\boldsymbol{\theta}^\star &= \arg\min_{\boldsymbol{\theta}} \mathrm{KL}\left[q(\mathbf{w}|\boldsymbol{\theta}) \| P(\mathbf{w}|\mathbf{y}, \mathbf{x})\right] \\
&= \arg\min_{\boldsymbol{\theta}} \int q(\mathbf{w}|\boldsymbol{\theta}) \log \frac{q(\mathbf{w}|\boldsymbol{\theta})}{P(\mathbf{w})P(\mathbf{y}|\mathbf{x}, \mathbf{w})} \, \mathrm{d}\mathbf{w} \qquad \text{(Apply Equation 3)} \\
&= \arg\min_{\boldsymbol{\theta}} -\mathbb{E}_{q(\mathbf{w}|\boldsymbol{\theta})}\left[\log P(\mathbf{y}|\mathbf{x}, \mathbf{w})\right] + \mathrm{KL}\left[q(\mathbf{w}|\boldsymbol{\theta}) \| P(\mathbf{w})\right].
\end{aligned} \tag{4}$$

We define the resulting cost function from Eq. (4) as:

$$\mathcal{L}(\mathbf{x}, \mathbf{y}) = \underbrace{-\mathbb{E}_{q(\mathbf{w}|\boldsymbol{\theta})}\left[\log P(\mathbf{y}|\mathbf{x}, \mathbf{w})\right]}_{\text{data-dependent term}} + \underbrace{\mathrm{KL}\left[q(\mathbf{w}|\boldsymbol{\theta}) \| P(\mathbf{w})\right]}_{\text{prior matching term}}. \tag{5}$$

The loss function $\mathcal{L}(\mathbf{x}, \mathbf{y})$ in Eq. (5), also known as the variational free energy, consists of two components: the prior matching term and the data-dependent term. The prior matching term can be approximated using the Monte Carlo method or computed analytically if a closed-form solution exists. The data-dependent term is equivalent to minimising the mean squared error between the input-output pairs in the training data. To optimise $\mathcal{L}(\mathbf{x}, \mathbf{y})$, the prior distribution $P(\mathbf{w})$ must be defined. In Sec. 3.2, we define a dynamic prior that accelerates convergence and better models complex one-to-many mappings in the data.

3.2 Momentum Prior with Exponential Moving Average

In our preliminary work, significant performance degradation is observed when using naive Gaussian (*e.g.*, $\mathcal{N}(\mathbf{0}, \mathbf{I})$) or empirical Bayes priors. To address this, we propose the *Momentum Prior*, a simple yet effective strategy that uses an exponential moving average to stabilise training by smoothing parameter updates and promoting convergence to better local optima. Suppose that the variational posterior is a diagonal Gaussian, then the variational posterior parameters are $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\sigma})$. A posterior sample of the weights $\mathbf{w}$ can be obtained via the reparameterisation trick (Kingma, 2014).

$$\mathbf{w} = \boldsymbol{\mu} + \boldsymbol{\sigma} \circ \boldsymbol{\epsilon} \quad \text{with } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{6}$$

Having liberated our algorithm from the confines of fixed priors, we propose a dynamic prior by updating the prior's parameters to the exponential moving average (EMA) of the variational posterior parameters. Specifically, for the prior $P(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \boldsymbol{\mu}_t^{\text{EMA}}, {\boldsymbol{\sigma}_t^{\text{EMA}}}^2\mathbf{I})$, the parameters are updated at each minibatch training step t over the training period $[0, 1, 2, \ldots, T]$ as follows:

$$\begin{aligned} \boldsymbol{\mu}_0^{\text{EMA}} &= \mathbf{0}, \;\; \boldsymbol{\sigma}_0^{\text{EMA}} = \sigma^{\text{o}}\mathbf{1}, \\ \boldsymbol{\mu}_t^{\text{EMA}} &= \beta\boldsymbol{\mu}_{t-1}^{\text{EMA}} + (1-\beta)\boldsymbol{\mu}_t, \quad t = 1...T, \\ \boldsymbol{\sigma}_t^{\text{EMA}} &= \beta\boldsymbol{\sigma}_{t-1}^{\text{EMA}} + (1-\beta)\boldsymbol{\sigma}_t, \quad t = 1...T, \end{aligned} \tag{7}$$

where $\boldsymbol{\mu}_t$ and $\boldsymbol{\sigma}_t$ represent the mean and variance from the variational posterior $q(\mathbf{w}|\boldsymbol{\theta})$ at training step t, σ^{o} is a scalar controlling the magnitude of initial variance in the prior distribution, and β denotes the EMA decay rate. Thereafter, for minibatch optimisation with M image pairs, we update $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\sigma})$ at step t by minimising minibatch loss $\mathcal{L}^{\text{mini}}(\mathbf{x}, \mathbf{y})$, reformulated from Eq. (5) as:

$$\begin{aligned} \mathcal{L}^{\text{mini}}(\mathbf{x}, \mathbf{y}) &= \underbrace{-\mathbb{E}_{q(\mathbf{w}|\boldsymbol{\theta})}\left[\log P(\mathbf{y}|\mathbf{x}, \mathbf{w})\right]}_{\text{data-dependent term}} + \underbrace{\frac{1}{M}\mathrm{KL}\left[q(\mathbf{w}|\boldsymbol{\theta}) \| P(\mathbf{w})\right]}_{\text{prior matching term}}, \\ &= \frac{1}{M}\Big[\underbrace{\sum_i^M \mathbb{E}_{\mathbf{w}\sim q(\mathbf{w}|\theta)}\|F(\mathbf{x}^i; \mathbf{w}) - \mathbf{y}^i\|_2^2}_{\text{data-dependent term}} + \underbrace{\log\frac{\boldsymbol{\sigma}_t^{\text{EMA}}}{\boldsymbol{\sigma}} + \frac{\boldsymbol{\sigma}^2 + (\boldsymbol{\mu} - \boldsymbol{\mu}_t^{\text{EMA}})^2}{2{\boldsymbol{\sigma}_t^{\text{EMA}}}^2} - \frac{1}{2}}_{\text{prior matching term}}\Big], \end{aligned} \tag{8}$$

where the prior matching term is expressed as the analytical solution of $\mathrm{KL}\left[q(\mathbf{w}|\boldsymbol{\theta}) \| P(\mathbf{w})\right]$.

The momentum prior is motivated by the following reasoning: it begins with a naive Gaussian prior early in training, offering useful inductive biases (Wilson & Izmailov, 2020). However, as training progresses, relying on a fixed simple prior can restrict the network's capacity to fit the data effectively. To overcome this, the momentum prior gradually updates its parameters with empirical information from the data during training. The momentum prior is akin to the momentum teacher (He et al., 2020; Grill et al., 2020) in self-supervised learning but differs by regularising variational posterior parameters instead of student model outputs. This simple idea significantly improves BNN performance on our task. Additionally, the momentum prior also shares similarities with deep learning ensembles (Lakshminarayanan et al., 2017), a key strategy for uncertainty estimation, as per Ashukha et al. (2020). Unlike empirical Bayes (Robbins, 1956; Krishnan et al., 2020), which defines a static prior based on MLE-optimised parameters, our momentum-based strategy incrementally refines the prior during training. This continuous adaptation prevents the model from exploiting shortcut learning when optimising the data-dependent term in Eq. (5), thereby avoiding sub-optimal solutions.

3.3 Predictions Under Uncertainty

After optimising the variational posterior parameters $\boldsymbol{\theta}^\star$ via Eq. (4), predictions are made by sampling weights $\mathbf{w}$ from the variational posterior distribution $q(\mathbf{w}|\boldsymbol{\theta})$. As shown in Algorithm 1, we sample K sets of network weights $\{\mathbf{w}_k\}_{k=1}^{K}$, where each $\mathbf{w}_k$ is used to produce a corresponding output $\hat{\mathbf{y}}_k$ via $F(\mathbf{x};\mathbf{w}_k)$. A quality metric D is then employed to rank the K candidates and select the most suitable output $\mathbf{y}^{\text{opt}}$, with higher D-values indicating better quality.

The prediction process is described for two cases depending on the availability of a reference:

i) With reference: When a reference image $\mathbf{y}$ is available, the quality metric D can be instantiated as the negative mean squared error (MSE) or other perceptual metrics to rank the K candidates, with the best score determining the final output.

ii) Without reference: in the absence of a reference image, the quality metric $D(\cdot)$ can be a no-reference image quality metric, such as negative NIQE (Mittal et al., 2012), UIQM (Panetta et al., 2015), or UCIQE (Yang & Sowmya, 2015). Alternatively, vision-language models like CLIP (Radford et al., 2021; Wang et al., 2023) can be used to find the best-matching image based on a given textual description. For instance, CLIP's encoders can extract features from a predicted image $\hat{\mathbf{y}}_k$ and a text prompt (e.g., "`A bright photo`"), denoted as $\mathbf{h}_k$ and $\mathbf{h}_{\text{text}}$, respectively. The quality metric D is then defined as their cosine similarity: $D(\hat{\mathbf{y}}_k) = \frac{\mathbf{h}_k^\top \mathbf{h}_{\text{text}}}{\|\mathbf{h}_k\|\|\mathbf{h}_{\text{text}}\|}$. We denote the BEM utilising CLIP as BEM_{CLIP}. Meanwhile, our BEM can perform deterministic predictions (i.e., without requiring multiple weight samples) by simply setting $\mathbf{w} = \mathbf{u}$. We refer to this deterministic mode as $\text{BEM}_{\text{Determ.}}$. However, due to its deterministic nature, $\text{BEM}_{\text{Determ.}}$, like any deterministic model, is inherently sub-optimal for capturing complex one-to-many mappings.

Algorithm 1: Prediction

Input: Input $\mathbf{x}$, network F
Initialisation: the best score $s^{\text{best}} \leftarrow 0$;
for $k \leftarrow 1$ **to** K **do**
  Sample $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
  $\mathbf{w}_k \leftarrow$ Calculate Eq. (6);
  $\hat{\mathbf{y}}_k = F(\mathbf{x};\mathbf{w}_k)$;
  if reference $\mathbf{y}$ exists **then**
    $s_k = D(\hat{\mathbf{y}}_k, \mathbf{y})$; `// reference`
  else
    $s_k = D(\hat{\mathbf{y}}_k)$; `// no-reference`
  if $s_k > s^{\text{best}}$ **then**
    Update $s^{\text{best}} \leftarrow s_k$;
    Set $\mathbf{y}^{\text{opt}} \leftarrow \hat{\mathbf{y}}_k$;
Output: Optimal prediction $\mathbf{y}^{\text{opt}}$.

4 BNN + DNN: A two-stage approach

Image data is inherently high-dimensional. While BNN can be directly applied to model high-dimensional image data, it compromises precision due to the complexity involved (see Appendix E for detailed analysis). To address this issue, we propose to use BEM to model the one-to-many mapping in a lower-dimensional feature representation of image. Then, we project the image features back to the original pixel space by a DNN.

4.1 The Framework

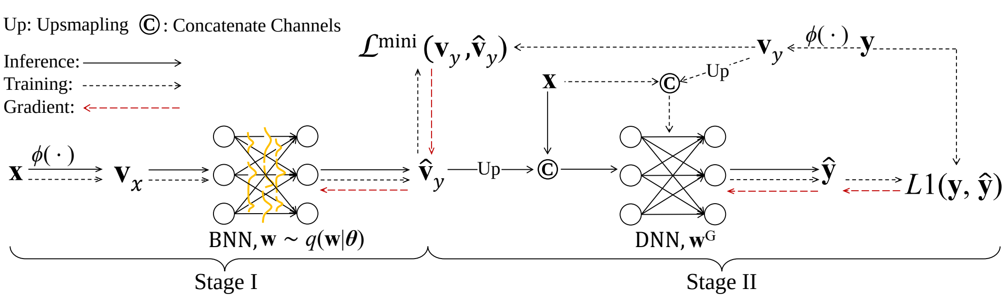


Figure 2: The two-stage pipeline. In Stage I, the BNN with weights $\mathbf{w} \sim q(\mathbf{w}|\boldsymbol{\theta})$ is trained by minimising the minibatch loss $\mathcal{L}^{\text{mini}}(\mathbf{v}_y, \hat{\mathbf{v}}_y)$ in Eq. (8). In Stage II, the DNN with weights $\mathbf{w}^{\text{G}}$ is trained by minimising the L1 loss, $L1(\mathbf{y}, \hat{\mathbf{y}})$. The inference process is denoted by $\rightarrow$, while the training process for each stage is indicated by $\dashrightarrow$. The gradient flow is shown with $\dashrightarrow$.

Figure 2 illustrates our proposed two-stage framework. We apply a reduction function ϕ to compress high-dimensional image data by either statistical summarisation or down-sampling, yielding compact representations $\mathbf{v}_x = \phi(\mathbf{x})$ and $\mathbf{v}_y = \phi(\mathbf{y})$ in a lower-dimensional space. In the first stage, the BEM models the complex one-to-many mapping between $\mathbf{v}_x$ and $\mathbf{v}_y$. In the second stage, a DNN G refines the results by taking the first-stage low-dimensional output $\hat{\mathbf{v}}_y$ along with the original low-quality image $\mathbf{x}$ as inputs, producing a high-quality recovered image. The overall process is formulated as:

$$\hat{\mathbf{v}}_y = F(\phi(\mathbf{x}); \mathbf{w}), \quad \mathbf{w} \sim q(\mathbf{w} \mid \boldsymbol{\theta}), \tag{9}$$

$$\hat{\mathbf{y}} = G(\hat{\mathbf{v}}_y, \mathbf{x}; \mathbf{w}^{\mathrm{G}}), \tag{10}$$

where $\mathbf{w}^{\mathrm{G}}$ denotes the weights of the second-stage model. We explore two reduction functions: bilinear downsampling and local 2D histogram. Both methods are effective; however, bilinear downsampling provides higher measurement values on full-reference image quality assessment metrics. Additionally, considering bilinear downsampling offers a more efficient computation, we adopt it as the default setting. Further analysis of the reduction function ϕ is provided in Appendix A.

During the training phase of the second-stage model, we use the downsampled features of the target image $\mathbf{y}$ along with the low-quality image $\mathbf{x}$ as input to the DNN, instead of using the output from the first-stage model. This strategy removes constraints imposed by the first-stage model, thereby allowing the second stage to reach its full potential. Importantly, as illustrated in the inference flow in Figure 2, the inference process remains independent of the target image. Further analysis for two-stage frameworks is provided in Appendix E.

Backbone Network. For both the first and the second stage models, we adopt the same backbone network, but with different input and output layers. To enable weight uncertainty for the first stage model, we convert all the convolution and linear layers in the backbone network to their Bayesian counterparts, the weight parameters of which are obtained via Eq. (6). Inspired by Mamba (Gu & Dao, 2023) and VMamba (Liu et al., 2024b), featuring their linear computational complexity for long sequence modelling, we employ a Mamba as the backbone of our BEM. The overall framework is akin to a U-Net. We provide the details and experiment with the backbone in Appendix B.

4.2 SPEEDING UP INFERENCE

Similar to diffusion models, our BEM benefits from multiple inference passes to produce high-quality outputs. However, unlike the sequential denoising process of diffusion models, BEM allows parallel execution. We accelerate inference using two main strategies: I) Applying Algorithm 1 only to the first-stage model to generate a low-resolution output, $\mathbf{v}^{\mathrm{opt}}$. With a $16\times$ downsampling in function ϕ, this provides a theoretical $256\times$ speedup. II) Parallelising the K iterations along the batch dimension achieves a speedup proportional to the GPU's parallel computing capability. As illustrated in Figure 3, the accelerated inference speed for image resolutions of 512^2 and 1024^2, is in the same level of the single-pass inference. However, when the function D does not support parallel execution, the speed decreases proportionally to D's computational complexity. This acceleration strategy introduces a minor degradation in image quality: at $K = 100$, we observe an average drop of 3.2% in PSNR, while no decrease is noted in UIQM.

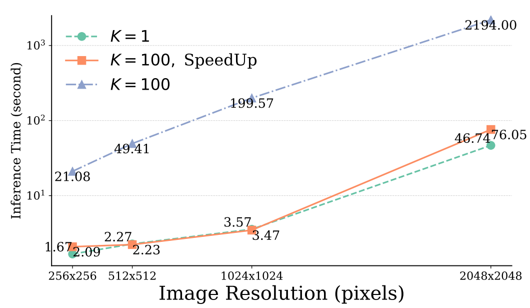


Figure 3: Inference speed before and after acceleration. A parallel implementation of D is employed. The model runs on an Nvidia RTX 4090.

5 EXPERIMENTS

Datasets. We conduct experiments on several low-light image enhancement (LLIE) and underwater image enhancement (UIE) datasets. For LLIE, we evaluate our method on LOL-v1 (Wei et al., 2018) and LOL-v2 (real and synthetic subsets)(Yang et al., 2021), both of which have training and test splits, as well as the unpaired LIME(Guo et al., 2016), NPE (Wang et al., 2013), MEF (Ma et al., 2015), DICM (Lee et al., 2013), and VV (Vonikakis et al., 2018) datasets. For UIE, we use the UIEB (Li et al., 2019a), U45 (Li et al., 2019b), and UCCS (Liu et al., 2020) datasets. The UIEB dataset is further divided into training, validation (R90), and test (C60) subsets.

Metrics. For paired datasets, we evaluate pixel-level accuracy using PSNR and SSIM, and perceptual quality using LPIPS (Zhang et al., 2018). For real-world datasets, we use NIQE Mittal et al. (2012) as a no-reference metric. In UIE tasks, we additionally evaluate image quality using UIQM (Panetta et al., 2015) and UCIQE (Yang & Sowmya, 2015).

Settings. All models are trained with the Adam optimiser, starting at a learning rate of 2×10^{-4} and decaying to 10^{-6} using a cosine annealing schedule. The first-stage model is trained for 300K iterations on inputs reduced to a size of 24×24 through function ϕ, while the second-stage model is trained for 150K iterations on inputs of size 128×128. Batch size M is set to 8, and ϕ defaults to bilinear downsampling with a $\frac{1}{16}$ scaling factor. Unless stated otherwise, K is 100, D in Algorithm 1 is negative MSE, and σ^o in Eq. (7) is set to 0.05.

5.1 Quantitative Results

Full-reference evaluation offers a limited view of model performance. Even without obvious distributional shifts between training and test sets, test results may not fully reflect the model's generalisation to real-world scenarios. In contrast, no-reference evaluation provides a more practical and meaningful measure of a model's utility in real-world applications.

Table 1: Full-reference evaluation on the LOL-v1 and v2 datasets. The BEM in grey selects the output based on the GT images. The best results are in **bold**, and the second-best are underlined.

Method	GT Mean	LOL-v1			LOL-v2-real			LOL-v2-syn		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
KinD (Zhang et al., 2019)	✗	19.66	0.820	0.156	18.06	0.825	0.151	17.41	0.806	0.255
Restormer (Zamir et al., 2022)	✗	22.43	0.823	0.147	18.60	0.789	0.232	21.41	0.830	0.144
SNR-Net (Xu et al., 2022)	✗	24.61	0.842	0.151	21.48	0.849	0.157	24.14	0.928	0.056
RetinexFormer (Cai et al., 2023)	✗	25.16	0.845	0.131	22.80	0.840	0.171	25.67	0.930	0.059
RetinexMamba (Bai et al., 2024)	✗	24.03	0.827	0.146	22.45	0.844	0.174	25.89	0.935	0.054
LLFlow (Wang et al., 2022)	✓	25.13	0.872	0.117	26.20	0.888	0.137	24.81	0.919	0.067
GlobalDiff (Hou et al., 2024)	✓	27.84	0.877	0.091	28.82	0.895	0.095	28.67	0.944	0.047
GLARE (Zhou et al., 2024)	✓	27.35	<u>0.883</u>	0.083	28.98	0.905	0.097	29.84	0.958	-
BEM (ours)	✗	26.83	0.877	0.072	28.89	0.902	0.076	29.22	0.955	0.031
BEM (ours)	✓	**28.80**	**0.884**	**0.069**	**32.66**	**0.915**	**0.060**	**32.95**	**0.964**	**0.026**
$\text{BEM}_{\text{Determ.}}$ (ours)	✓	28.30	0.881	0.072	<u>31.41</u>	<u>0.912</u>	<u>0.064</u>	30.58	0.958	0.033
BEM_{CLIP} (ours)	✓	<u>28.43</u>	0.882	<u>0.071</u>	30.01	0.910	0.076	<u>31.51</u>	<u>0.961</u>	<u>0.030</u>

Full-Reference Evaluation. For the LLIE tasks, we present quantitative comparisons with state-of-the-art methods on the LOL-v1 and LOL-v2 datasets, as detailed in Table 1. Our BEM significantly outperforms all previous methods across all metrics. Notably, on LOL-v2-real, BEM achieves an exceptionally high PSNR of 32.66 dB. Although deterministic models are considered sub-optimal in the one-to-many mapping problem, our $\text{BEM}_{\text{Determ.}}$ (deterministic mode) still surpasses the previous methods across all benchmarks. We observed that previous methods often struggle to maintain high perceptual quality (measured by LPIPS) while ensuring pixel-level accuracy. However, our BEM excels in both, delivering the highest SSIM (0.877) and the lowest LPIPS (0.072). For the UIE

Table 2: Quantitative comparisons on the UIEB-R90, UIEB-C60, U45, and UCCS datasets in terms of PSNR, SSIM, UIQM, and UCIQE. Best results are in **bold**, second best are underlined.

Method	UIEB-R90		UIEB-C60		U45		UCCS	
	PSNR ↑	SSIM ↑	UIQM ↑	UCIQE ↑	UIQM ↑	UCIQE ↑	UIQM ↑	UCIQE ↑
WaterNet (Li et al., 2019a)	21.04	0.860	2.399	**0.591**	-	-	2.275	0.556
Ucolor (Li et al., 2021)	20.13	0.877	2.482	0.553	3.148	0.586	3.019	0.550
PUIE-MP (Fu et al., 2022)	21.05	0.854	2.524	0.561	3.169	0.569	2.758	0.489
Restormer (Zamir et al., 2022)	23.82	0.903	2.688	0.572	3.097	0.600	2.981	0.542
CECF (Cong et al., 2024)	21.82	0.894	-	-	-	-	-	-
FUnIEGAN (Islam et al., 2020)	19.12	0.832	2.867	0.556	2.495	0.545	3.095	0.529
PUGAN (Cong et al., 2023)	22.65	0.902	2.652	0.566	-	-	2.977	0.536
U-Shape (Peng et al., 2023)	20.39	0.803	2.730	0.560	3.151	0.592	-	-
Semi-UIR (Huang et al., 2023)	22.79	0.909	2.667	<u>0.574</u>	3.185	0.606	3.079	0.554
WFI2-Net (Zhao et al., 2024a)	23.86	0.873	-	-	3.181	<u>0.619</u>	-	-
BEM_{CLIP} (ours)	<u>24.36</u>	<u>0.921</u>	<u>2.885</u>	0.554	<u>3.266</u>	0.608	<u>3.115</u>	<u>0.558</u>
BEM (ours)	**25.62**	**0.940**	**2.931**	0.567	**3.406**	**0.620**	**3.224**	**0.561**

tasks, we present quantitative comparisons on the UIEB-R90 dataset, as shown in Table 2. Our BEM outperforms the second-best WFI2-Net by 1.76 dB in PSNR. This superior performance, observed consistently across both LLIE and UIE tasks, highlights BEM's effectiveness and versatility.

No-Reference Evaluation. For no-reference low-light images, we recover them using Algorithm 1 and D is instantiated as the NIQE metric. We then evaluate our method on five unpaired datasets as shown in Table 3, where we report the NIQE scores of SOTA methods. Our BEM consistently outperforms prior methods across all datasets. For enhancing no-reference underwater images, we instantiate D in Algorithm 1 as the UIQM and UCIQE metrics. We then evaluate our method on the C60, U45 and UCCS test sets. As shown in Table 2, BEM achieves the best UIQM scores across all test sets. With the UCIQE metric, we also achieve the best results in the U45 and UCCS test sets. These results, spanning different tasks and datasets, demonstrate the robustness and effectiveness of our method in real-world applications.

Table 3: No-reference evaluation on LIME, NPE, MEF, DICM and VV, in terms of NIQE↓. The best results are in **blodface**.

Method	DICM	LIME	MEF	NPE	VV
KinD (Zhang et al., 2019)	5.15	5.03	5.47	4.98	4.30
ZeroDCE (Guo et al., 2020)	4.58	5.82	4.93	4.53	4.81
RUAS (Liu et al., 2021)	5.21	4.26	3.83	5.53	4.29
LLFlow (Wang et al., 2022)	4.06	4.59	4.70	4.67	4.04
PairLIE (Fu et al., 2023b)	4.03	4.58	4.06	4.18	3.57
RFR (Fu et al., 2023a)	3.75	3.81	3.92	4.13	-
GLARE (Zhou et al., 2024)	3.61	4.52	3.66	4.19	-
CIDNet (Feng et al., 2024)	3.79	4.13	3.56	3.74	3.21
$\text{BEM}_{\text{Determ.}}$ (ours)	3.77	3.94	3.22	3.85	2.95
BEM (ours)	**3.55**	**3.56**	**3.14**	**3.72**	**2.91**

5.2 Visual Analysis

Predictions of One-to-Many. In Figure 4, we visualise the prediction process of BEM, where multiple plausible candidates are generated. As shown at the top of the figure, these candidates exhibit apparent visual differences. The best prediction candidate is selected using Algorithm 1, which is visually closer to the reference image. For no-reference prediction, we demonstrate that using the CLIP score with the text prompt, "`A bright photo`", results in the brightest image being outputted. By instantiating D as the NIQE metric, we can avoid generating overexposed predictions, as shown at the bottom right.

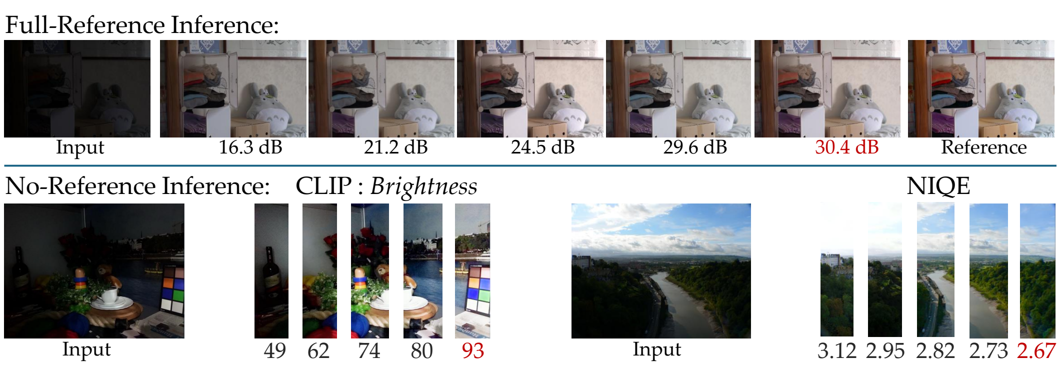


Figure 4: Visualisation of the predicting process of BEM in both cases with reference (top) and without reference (bottom). Zoom in for more details.

Qualitative Comparisons. We visually compare our BEM with twelve state-of-the-art UIE methods, including WaterNet (Li et al., 2019a), PRWNet (Huo et al., 2021), FUnIEGAN (Islam et al., 2020), PUGAN Cong et al. (2023), MMLE (Zhang et al., 2022), PUIE-MP (Fu et al., 2022), FiveA+(Jiang et al., 2023b), CLUIE (Li et al., 2023), Semi-UIR (Huang et al., 2023), UColor (Li et al., 2021), DM-Underwater (Tang et al., 2023), and CLIP-UIE (Liu et al., 2024a). As depicted in the first and second rows of Figure 5, our BEM achieves superior removal of underwater turbidity compared to other methods. In deeper ocean images with dominant blueish effects (last row in Figure 5), BEM can better enhance visual clarity. Visual comparisons on five unpaired LLIE test sets are shown in Figure 6, where our restored images offer better perceptual improvement. For example, in DICM, our method enhances brightness while effectively avoiding overexposure. These visual improvements align with the superior quantitative results presented in Sec. 5.1. HD visual results are included in Appendix E.

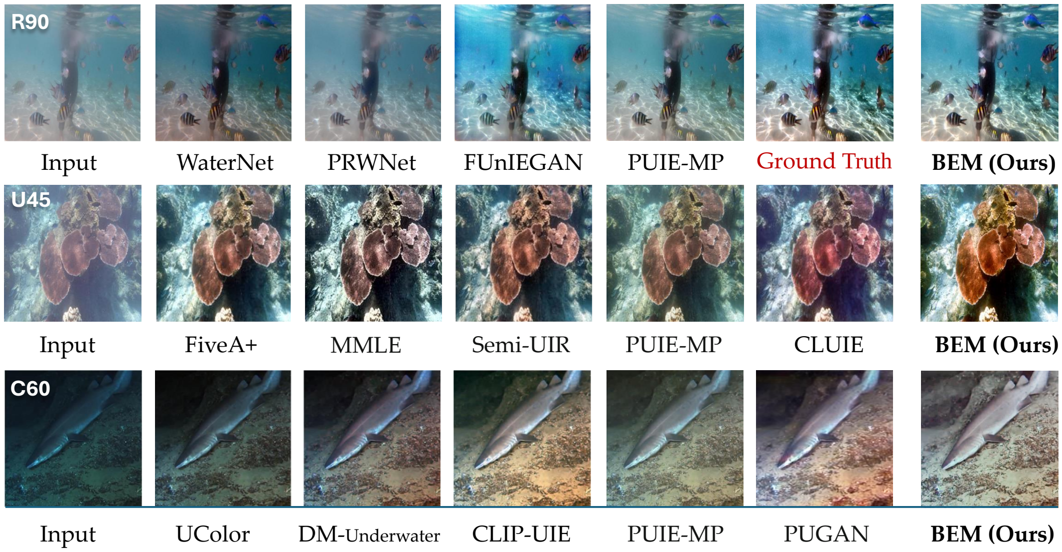


Figure 5: Visual comparisons on the R90, C60 and U45 datasets. Best viewed when zoomed in.

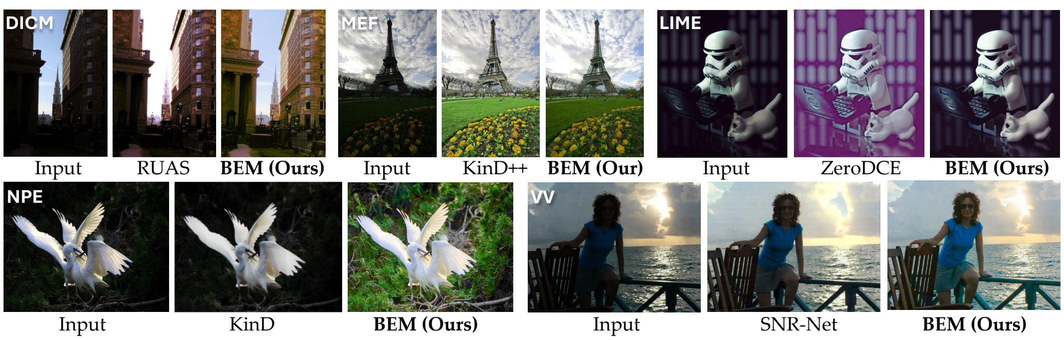


Figure 6: Visual comparisons on the DICM, LIME, MEF, NPE and VV datasets.

5.3 Ablation Studies

Single-Stage *vs*. Two-Stage Approaches. We assess the performance of our two-stage approach by comparing it against a single-stage variant. As discussed in Sec. 4, directly converting a DNN into a BNN typically results in noisy predictions. To generate smooth outputs, our single-stage model retains the last layer in the network as a deterministic layer, the entire process of which is opposite to the Bayesian last layer method (Harrison et al., 2024). While the two-stage approach introduces only marginal additional computational overhead, its performance significantly surpasses that of the single-stage model, as shown in Table 4. This highlights the efficiency and effectiveness of our two-stage approach.

Table 4: Single-stage *vs*. two-stage approaches on LOL-v1. FLOPs are calculated in an input size of 256×256 pixels.

Model	FLOPs	PSNR ↑	SSIM ↑
Single Stage	20.41G	24.78	0.852
Two Stages	20.49G	26.83	0.877

Magnitude of Uncertainty. The performance improvements of our BEM primarily stem from its ability to effectively model the one-to-many mapping using BNNs. To support this claim, we evaluate the influence of the variance in the variational posterior on model performance. As shown in

Figure 7, except for BEM with $\sigma^\circ = 0.0001$, all other BEM instances outperform the DNN. This indicates that by setting a moderate variance in the momentum prior, BEM can significantly surpass its DNN counterpart.

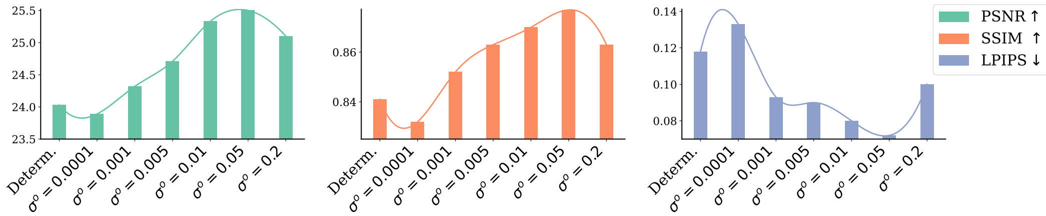


Figure 7: Effect of initial variance values (*i.e.*, σ^{o} in Eq. 7) on model performance. The results are obtained by evaluating single-stage models on the LOL-v1 dataset. "Determ." denotes the deterministic baseline model.

Impact of Different Priors. We evaluate the effectiveness of our momentum prior against two baseline priors: a naive Gaussian prior and an empirical Bayes prior. The naive Gaussian prior is defined as $P(\mathbf{W}) = \mathcal{N}(\mathbf{0}, 0.1\mathbf{I})$. The empirical Bayes prior, MOPED (Krishnan et al., 2020), is defined as $P(\mathbf{W}) = \mathcal{N}(\mathbf{w}^{\text{MLE}}, 0.1\mathbf{I})$, where $\mathbf{w}^{\text{MLE}}$ represents the maximum likelihood estimate (MLE) of the weights learned by optimising a deterministic network. In the case of the empirical Bayes prior, the mean $\boldsymbol{\mu}$ of the variational posterior $q(\mathbf{w}|\boldsymbol{\theta})$ is initialised as the MLE of the weights, $\mathbf{w}^{\text{MLE}}$, and the posterior variance $\boldsymbol{\sigma}$ is set to $0.1\mathbf{w}^{\text{MLE}}$, as suggested by Krishnan et al. (2020). As shown in Figure 8, the momentum prior demonstrates a clear advantage over both baselines. While the empirical Bayes prior accelerates training during early iterations, its performance degrades over time due to the fixed nature of the prior. The fixed prior, learned from the same data, can act as a shortcut during the optimisation of the variational posterior parameters, minimising the loss function in Eq. (5) predominantly by reducing the prior matching term $\text{KL}\left[q(\mathbf{w}|\boldsymbol{\theta})\|P(\mathbf{w})\right]$. This behaviour bypasses data-driven learning, ultimately resulting in sub-optimal solutions that do not fully capture the data's inherent uncertainty.

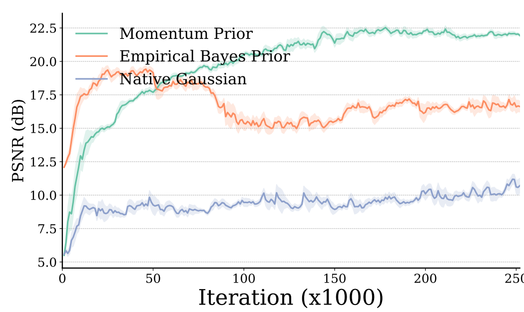


Figure 8: Training curves of one-stage BEMs with different priors. The PSNR for each iteration is calculated using the mean weight $\boldsymbol{\mu}$.

6 Discussion and Conclusion

Although BEM demonstrates stronger generalisation capability than DNN-based methods, fully realising its potential will require intentionally collecting target images under diverse capture settings to further increase label diversity. While using small image crops as training data can alleviate the label diversity problem to some extent, similar to conventional data augmentation strategies in DNNs, this approach has limitations. We leave these aspects for future work. Additionally, the distinction between image enhancement and image restoration is not always well-defined, as some restoration tasks (e.g., image colourisation and de-raining) may also present one-to-many mapping challenges. Consequently, our BEM could be extended to certain image restoration scenarios.

Overall, we identified the one-to-many mapping problem as a key limitation in existing image enhancement tasks and introduced the first Bayesian-based model to address this issue. To facilitate efficient training on high-dimensional data, we proposed a *Momentum Prior* that dynamically refines the prior distribution during training, enhancing convergence and performance. Our two-stage framework integrates the strengths of BNNs and DNNs, yielding a flexible yet computationally efficient model. Extensive experiments on various image enhancement benchmarks demonstrate significant performance gains over state-of-the-art models, showcasing the potential of Bayesian probabilistic models in handling the inherent ambiguities of image enhancement tasks, paving the way for future research in modelling complex one-to-many mappings in low-level vision tasks.

REFERENCES

Nantheera Anantrasirichai and David Bull. Contextual colorization and denoising for low-light ultra high resolution sequences. In *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 1614–1618. IEEE, 2021.

Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. *International Conference on Learning Representations (ICLR)*, 2020.

Jiesong Bai, Yuhao Yin, and Qiyuan He. Retinexmamba: Retinex-based mamba for low-light image enhancement. *arXiv preprint arXiv:2405.03349*, 2024.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

Yuanhao Cai, Hao Bian, Jing Lin, Haoqian Wang, Radu Timofte, and Yulun Zhang. Retinexformer: One-stage retinex-based transformer for low-light image enhancement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12504–12513, 2023.

Runmin Cong, Wenyu Yang, Wei Zhang, Chongyi Li, Chun-Le Guo, Qingming Huang, and Sam Kwong. Pugan: Physical model-guided underwater image enhancement using gan with dual-discriminators. *IEEE Transactions on Image Processing*, 32:4472–4485, 2023.

Xiaofeng Cong, Jie Gui, and Junming Hou. Underwater organism color fine-tuning via decomposition and guidance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 1389–1398, 2024.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Michael Dusenberry, Ghassen Jerfel, Yeming Wen, Yian Ma, Jasper Snoek, Katherine Heller, Balaji Lakshminarayanan, and Dustin Tran. Efficient and scalable bayesian neural nets with rank-1 factors. In *International conference on machine learning*, pp. 2782–2792. PMLR, 2020.

Cameron Fabbri, Md Jahidul Islam, and Junaed Sattar. Enhancing underwater imagery using generative adversarial networks. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 7159–7165. IEEE, 2018.

Yixu Feng, Cheng Zhang, Pei Wang, Peng Wu, Qingsen Yan, and Yanning Zhang. You only need one color space: An efficient network for low-light image enhancement. *arXiv preprint arXiv:2402.05809*, 2024.

Huiyuan Fu, Wenkai Zheng, Xiangyu Meng, Xin Wang, Chuanming Wang, and Huadong Ma. You do not need additional priors or regularizers in retinex-based low-light image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18125–18134, 2023a.

Zhenqi Fu, Wu Wang, Yue Huang, Xinghao Ding, and Kai-Kuang Ma. Uncertainty inspired underwater image enhancement. In *European conference on computer vision*, pp. 465–482. Springer, 2022.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22252–22261, 2023b.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Alex Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1780–1789, 2020.

Xiaojie Guo, Yu Li, and Haibin Ling. Lime: Low-light image enhancement via illumination map estimation. *IEEE Transactions on image processing*, 26(2):982–993, 2016.

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*, 90:103712, 2023.

James Harrison, John Willes, and Jasper Snoek. Variational bayesian last layers. In *International Conference on Learning Representations (ICLR)*, 2024.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pp. 5–13, 1993.

Jinhui Hou, Zhiyu Zhu, Junhui Hou, Hui Liu, Huanqiang Zeng, and Hui Yuan. Global structure-aware diffusion process for low-light image enhancement. *Advances in Neural Information Processing Systems*, 36, 2024.

Shirui Huang, Keyan Wang, Huan Liu, Jun Chen, and Yunsong Li. Contrastive semi-supervised learning for underwater image restoration via reliable bank. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18145–18155, 2023.

Fushuo Huo, Bingheng Li, and Xuegui Zhu. Efficient wavelet boost learning-based multi-stage progressive refinement network for underwater image enhancement. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1944–1952, 2021.

Md Jahidul Islam, Youya Xia, and Junaed Sattar. Fast underwater image enhancement for improved visual perception. *IEEE Robotics and Automation Letters*, 5(2):3227–3234, 2020.

Hai Jiang et al. Low-light image enhancement with wavelet-based diffusion models. *ACM Transactions on Graphics (TOG)*, 42(6):1–14, 2023a.

Jingxia Jiang, Tian Ye, Jinbin Bai, Sixiang Chen, Wenhao Chai, Shi Jun, Yun Liu, and Erkang Chen. Five a^+ network: You only need 9k parameters for underwater image enhancement. *British Machine Vision Conference (BMVC)*, 2023b.

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*, 30:2340–2349, 2021.

Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *2016 IEEE international conference on Robotics and Automation (ICRA)*, pp. 4762–4769. IEEE, 2016.

Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491, 2018.

Diederik P Kingma. Auto-encoding variational bayes. *International Conference on Learning Representations (ICLR)*, 2014. URL `http://arxiv.org/abs/1312.6114`.

Ranganath Krishnan, Mahesh Subedar, and Omesh Tickoo. Specifying weight priors in bayesian deep neural networks with empirical bayes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 4477–4484, 2020.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Chulwoo Lee, Chul Lee, and Chang-Su Kim. Contrast enhancement based on layered difference representation of 2d histograms. *IEEE transactions on image processing*, 22(12):5372–5384, 2013.

Chongyi Li, Chunle Guo, Wenqi Ren, Runmin Cong, Junhui Hou, Sam Kwong, and Dacheng Tao. An underwater image enhancement benchmark dataset and beyond. *IEEE transactions on image processing*, 29:4376–4389, 2019a.

Chongyi Li, Saeed Anwar, Junhui Hou, Runmin Cong, Chunle Guo, and Wenqi Ren. Underwater image enhancement via medium transmission-guided multi-color space embedding. *IEEE Transactions on Image Processing*, 30:4985–5000, 2021.

Hanyu Li, Jingjing Li, and Wei Wang. A fusion adversarial underwater image enhancement network with a public test dataset. *arXiv preprint arXiv:1906.06819*, 2019b.

Kunqian Li, Li Wu, Qi Qi, Wenjie Liu, Xiang Gao, Liqin Zhou, and Dalei Song. Beyond single reference for training: Underwater image enhancement via comparative learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(6):2561–2576, 2023.

Risheng Liu, Xin Fan, Ming Zhu, Minjun Hou, and Zhongxuan Luo. Real-world underwater enhancement: Challenges, benchmarks, and solutions under natural light. *IEEE transactions on circuits and systems for video technology*, 30(12):4861–4875, 2020.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10561–10570, 2021.

Shuaixin Liu, Kunqian Li, and Yilin Ding. Underwater image enhancement by diffusion model with customized clip-classifier. *arXiv preprint arXiv:2405.16214*, 2024a.

Yue Liu et al. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*, 2024b.

Kede Ma, Kai Zeng, and Zhou Wang. Perceptual quality assessment for multi-exposure image fusion. *IEEE Transactions on Image Processing*, 24(11):3345–3356, 2015.

Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal processing letters*, 20(3):209–212, 2012.

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Karen Panetta, Chen Gao, and Sos Agaian. Human-visual-system-inspired underwater image quality measures. *IEEE Journal of Oceanic Engineering*, 41(3):541–551, 2015.

Tongyao Pang, Yuhui Quan, and Hui Ji. Self-supervised bayesian deep learning for image recovery with applications to compressive sensing. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pp. 475–491. Springer, 2020.

Lintao Peng, Chunli Zhu, and Liheng Bian. U-shape transformer for underwater image enhancement. *IEEE Transactions on Image Processing*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Ali M Reza. Realization of the contrast limited adaptive histogram equalization (clahe) for real-time image enhancement. *Journal of VLSI signal processing systems for signal, image and video technology*, 38:35–44, 2004.

Herbert Robbins. An empirical bayes approach to statistics. *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, 1:157–163, 1956.

Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1874–1883, 2016.

Yi Tang, Hiroshi Kawasaki, and Takafumi Iwaguchi. Underwater image enhancement by transformer-based diffusion model with non-uniform sampling for skip strategy. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 5419–5427, 2023.

Marcin Tomczak, Siddharth Swaroop, Andrew Foong, and Richard Turner. Collapsed variational bounds for bayesian neural networks. *Advances in Neural Information Processing Systems*, 34: 25412–25426, 2021.

Vassilios Vonikakis, Rigas Kouskouridas, and Antonios Gasteratos. On the evaluation of illumination compensation algorithms. *Multimedia Tools and Applications*, 77:9211–9231, 2018.

Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 2555–2563, 2023.

Shuhang Wang, Jin Zheng, Hai-Miao Hu, and Bo Li. Naturalness preserved enhancement algorithm for non-uniform illumination images. *IEEE transactions on image processing*, 22(9):3538–3548, 2013.

Yudong Wang, Jichang Guo, Huan Gao, and Huihui Yue. Uiecˆ 2-net: Cnn-based underwater image enhancement using two color space. *Signal Processing: Image Communication*, 96:116250, 2021.

Yufei Wang, Renjie Wan, Wenhan Yang, Haoliang Li, Lap-Pui Chau, and Alex Kot. Low-light image enhancement with normalizing flow. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 2604–2612, 2022.

Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. *British Machine Vision Conference (BMVC)*, 2018.

Andrew G Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *Advances in neural information processing systems*, 33:4697–4708, 2020.

Xiaogang Xu, Ruixing Wang, Chi-Wing Fu, and Jiaya Jia. Snr-aware low-light image enhancement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17714–17724, 2022.

Miao Yang and Arcot Sowmya. An underwater color image quality evaluation metric. *IEEE Transactions on Image Processing*, 24(12):6062–6071, 2015.

Wenhan Yang, Wenjing Wang, Haofeng Huang, Shiqi Wang, and Jiaying Liu. Sparse gradient regularized deep retinex network for robust low-light image enhancement. *IEEE Transactions on Image Processing*, 30:2072–2086, 2021.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5728–5739, 2022.

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*, pp. 586–595, 2018.

Weidong Zhang, Peixian Zhuang, Hai-Han Sun, Guohou Li, Sam Kwong, and Chongyi Li. Underwater image enhancement via minimal color loss and locally adaptive contrast enhancement. *IEEE Transactions on Image Processing*, 31:3997–4010, 2022.

Yonghua Zhang, Jiawan Zhang, and Xiaojie Guo. Kindling the darkness: A practical low-light image enhancer. In *Proceedings of the 27th ACM international conference on multimedia*, pp. 1632–1640, 2019.

Chen Zhao, Weiling Cai, Chenyu Dong, and Chengwei Hu. Wavelet-based fourier information interaction with frequency diffusion adjustment for underwater image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8281–8291, 2024a.

Chen Zhao, Chenyu Dong, and Weiling Cai. Learning a physical-aware diffusion model based on transformer for underwater image enhancement. *arXiv preprint arXiv:2403.01497*, 2024b.

Han Zhou, Wei Dong, Xiaohong Liu, Shuaicheng Liu, Xiongkuo Min, Guangtao Zhai, and Jun Chen. Glare: Low light image enhancement via generative latent feature based codebook retrieval. *Proceedings of the European conference on computer vision (ECCV)*, 2024.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

A Experiments on Reduction Function ϕ

Regarding the form of reduction function ϕ in Eq. (9). we consider two instantiations: bilinear downsampling and local 2D histogram. As illustrated in Figure 9. with the local histogram, the recovered images preserve more details than that of bilinear downsampling, due to additional configuration for the histogram's bin number, avoiding losing much information when the downsample scale is larger.

(a) Bilinear Down (b) Local Histogram

Figure 9: With the same downsampling scale, the local histogram offers more precise control over the amount of retained information by adjusting the number of bins (corresponding to the number of channels). In contrast, bilinear downsampling tends to lose excessive details, especially when using larger downsampling strides.

The discrete nature of histograms poses challenges in both prediction accuracy and computational speed. To address this, we approximate the histogram calculation using Kernel Density Estimation (KDE), which significantly improves both computation efficiency and prediction accuracy. As shown in Table 5, while the pixel-level PSNR of local histogram-based ϕ is slightly lower than that of bilinear downsampling, we attribute this to the larger variance inherent in histogram values, which the model struggles to fit effectively.

Table 5: Comparisons of different instantiations of ϕ. The PSNR values on LOL-v1 are reported. K is set to 100.

Function ϕ	Downscale	Bins	Channels	PSNR $\uparrow$
Bilinear Down	8	N/A	3	25.87
Local Histogram	8	3	9	25.29
Local Histogram	8	10	30	24.96
Local Histogram	8	16	48	24.80
Bilinear Down	16	N/A	3	26.83
Local Histogram	16	10	30	25.89
Local Histogram	16	16	48	25.83

Despite this, we observe that the local histogram approach exhibits slightly better colour representation compared to the bilinear instance. In Figure 10, we present a visual comparison between the two implementations, highlighting that the histogram-based model generates more vivid colours. However, the bilinear downsampling method performs better in restoring details in areas where significant information loss occurs.

Figure 10: Visual comparison between the local histogram and bilinear downsampling implementations of the reduction function ϕ. The bilinear ϕ demonstrates better restoration capability compared to the histogram-based counterpart. However, the histogram-based ϕ shows better global colour representation. Best viewed when zoomed in.

B Investigation on Mamba Backbone

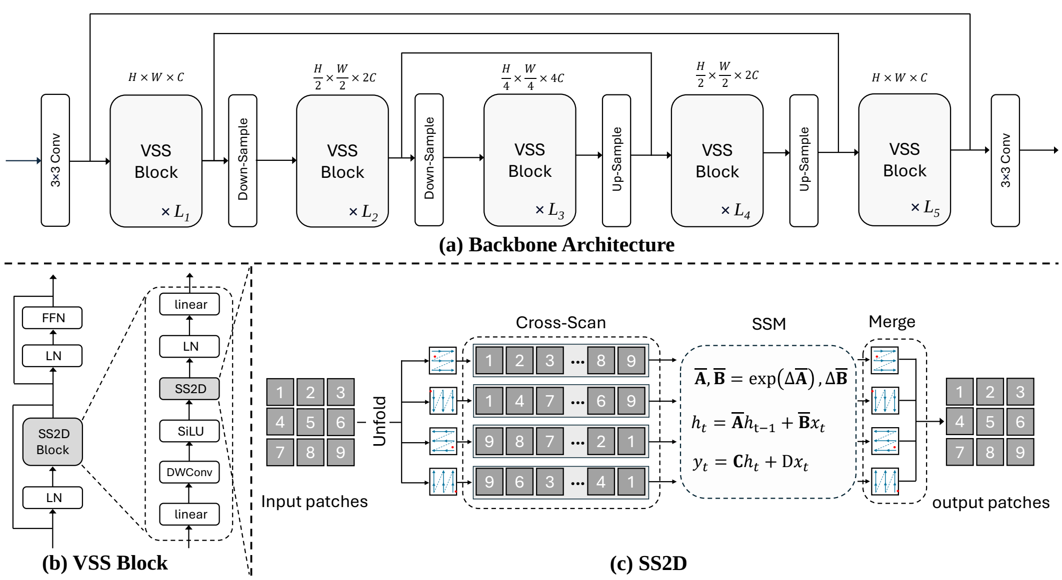


Figure 11: Overview of the Mamba backbone architecture, consisting of five feature stages, each comprising L_i VSS blocks. The shortcut connections are implemented using addition. Panel (a) illustrates the hierarchical structure of the backbone. Panel (b) details the VSS Block, including its integration with the SS2D module. Panel (c) explains the SS2D mechanism, incorporating Cross-Scan, structured state-space modelling (SSM), and patch merging. Further details about SS2D can be found in Liu et al. (2024b).

Considering Mamba's linear computational complexity for long sequence modelling, we adopt the VMmaba Liu et al. (2024b) to build the backbone of our BEM. The overall framework is akin to a

U-Net, but we replace all the Transformer blocks Dosovitskiy et al. (2020) with the Visual State-Space (VSS) blocks, each of which is composed of a 2D Selective Scan (SS2D) module Liu et al. (2024b) and a feedforward network (FFN). The formulation of VSS block Liu et al. (2024b) in layer l can be expressed as

$$\begin{aligned} \mathbf{h}_l &= \text{SS2D}\left(\text{LN}\left(\mathbf{h}_{l-1}\right)\right) + \mathbf{h}_{l-1}, \\ \mathbf{h}_{l+1} &= \text{FFN}\left(\text{LN}\left(\mathbf{h}_l\right)\right) + \mathbf{h}_l, \end{aligned} \tag{11}$$

where FFN denotes the feedforward network and LN denotes layer normalisation. $\mathbf{h}_{l-1}$ and $\mathbf{h}_l$ denote the input and output in the l-th layer, respectively. As shown in Figure 11, the Mamba backbone consists of an input convolutional layer, $L_1 + L_2 + L_3 + L_4 + L_5$ VSS blocks, and an output convolutional layer. After each downsampling operation, the spatial dimensions of the feature maps are halved, while the number of channels is doubled. Specifically, given an input image with a shape of $H \times W \times 3$, the encoding blocks obtain hierarchical feature maps of sizes $H \times W \times C$, $\frac{H}{2} \times \frac{W}{2} \times 2C$ and $\frac{H}{4} \times \frac{W}{4} \times 4C$. In the last two feature stages, the features are upsampled with the `pixelshuffle` layers (Shi et al., 2016). At each scale level, lateral connections are built to link the corresponding blocks in the encoder and decoder.

Construct the backbone. We build our backbone by gradually evaluating each configuration of a vanilla Mamaba-based UNet. We thoroughly investigate settings including `ssm-ratio`, block numbers, `n_feat` and `mlp-ratio`. The training strategies for all variants are identical. Setting `n_feat` denotes the number of feature maps in the first `conv3×3`'s output. Setting `d_state` denotes the state dimension of SSM. Note that the established baseline assures two things: 1) Further naively introducing additional parameters and FLOPs, e.g., scaling models with more blocks, will not help boost the performance. 2) A technique with additional parameters introduced to the baseline model can no doubt demonstrate its effectiveness if the modified model shows better results than the baseline.

Table 6: The performance of deterministic Mamba UNet variants with different `d_state`, `ssm-ratio`, `mlp-ratio`, `n_feat` and `block numbers`. PSNR and SSIM on LOL-v1 are reported. Since the deterministic networks trained using minibatch optimisation are likely to fit very different targets each time, the results will fluctuate greatly. We train each model five times and report the average performance.

d_state	ssm-ratio	mlp-ratio	n_feat	block numbers	FLOPs (G)	Params (M)	TP img/s	PSNR (dB)	SSIM
1	1	2.66	40	[2,2,2]	14.25	1.23	125	22.45	0.828
1	1	4	40	[2,2,2]	20.41	1.52	78	23.76	0.842
16	1	2.66	40	[2,2,2]	25.50	1.37	84	23.83	0.840
32	1	2.66	40	[2,2,2]	37.49	1.52	61	21.93	0.812
16	2	4	40	[2,2,2]	44.36	2.08	58	23.67	0.830
16	2	4	52	[2,2,2]	65.10	3.37	40	23.21	0.833
16	2	4	40	[2,2,2,2]	54.82	7.77	51	23.44	0.838
1	2	4	40	[2,2,2]	21.87	1.79	82	22.73	0.834

To balance both speed and performance, we selected the model in the second row of Table 6 as the backbone for our BEM. The chosen backbone features a simple architecture with no task-specific modules, enhancing its generalisability and establishing a solid foundation for extending our method to other types of vision tasks.

C Controllable Local Enhancement

Thanks to the interpretability of the lower-dimensional representations in both the spatial and channel dimensions, we can easily achieve local adjustment with a masking strategy. The local adjustment is particularly useful in the cases where the input images are unevenly distorted, and we want to retain the undistorted regions consistent before and after enhancement. The local adjustment process can be achieved by using a mask layer $\mathbf{M}$: $\mathbf{y}^{\text{local}} = G(\gamma\mathbf{M} \odot \mathbf{v}, \mathbf{x}; \mathbf{w}^{\text{G}})$, where $\mathbf{v}$ can be lower-dimensional features extracted from a real image or estimated by the first stage model via Eq. (9). We can use a scalar γ to control the strength of the enhancement effect. A demonstration of the local enchantment is shown in Figure 12.

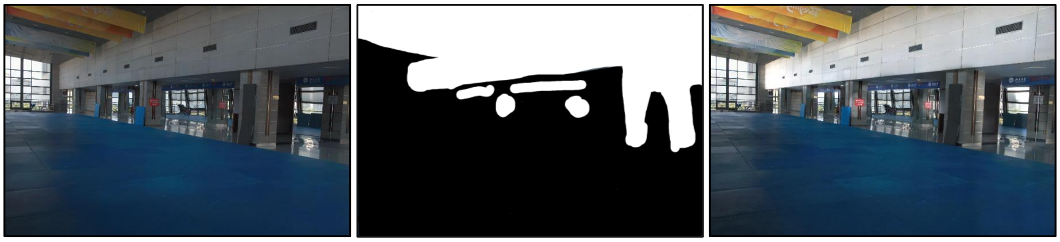


Figure 12: The local brightness of an image before adjustment (left) can be edited locally by providing a mask layer (middle). The image after adjustment (right) shows improved brightness in the regions indicated by the mask.

Compared to directly applying the mask to the output, our local enhancement strategy not only reduces the dependency on mask accuracy but also results in smoother transitions at the mask boundaries. This mitigates issues such as excessive roughness or colour inconsistencies between processed and unprocessed regions.

D Label Diversity Augmentation

Theoretically, an infinite number of target images could correspond to a single input. However, current paired datasets often lack sufficient label diversity, which may become a bottleneck for BEM model performance.

Table 7: Evaluation of label augmentation strategies for enhancing label diversity. PSNR scores are obtained using single-stage models on LOL-v1.

Model	Gamma Correction	Saturation Shift	CLAHE	PSNR ↑
BEM				24.78
BEM	✓			24.89
BEM	✓	✓		24.93
BEM	✓	✓	✓	24.86
DNN				24.02
DNN	✓	✓	✓	21.58

Without relying on additional data collection to increase label diversity, we propose two strategies for augmenting label diversity within existing datasets:

i) When training a deep network, high-resolution images are often divided into smaller crops (e.g., 128×128). Many of these smaller image crops may represent the same scene, but due to various factors, such as being captured at different moments in a video or having different capture settings, the corresponding target crops show differences in colour or brightness. Thus, using these crops as input during training, the actual label diversity within the training data is naturally increased.

ii) Existing labels can be further enriched by applying data augmentation techniques such as random brightness adjustments, saturation shifts, changes in colour temperature, gamma corrections, and histogram equalisation.

Both strategies contribute to increasing label diversity to some extent.

In Table 7, we evaluate whether expanding the number of target images using gamma correction, saturation shift, and CLAHE Reza (2004) can further improve the model's performance. Among these, saturation shift is a linear transformation, while gamma correction and CLAHE are nonlinear methods. We observed that deterministic networks showed a decline in performance after applying these label augmentation techniques. This can be attributed to DNNs overfitting to local solutions that deviate further from the inference image as uncertainty in the data increases. In contrast, BEM exhibited a slight increase in PSNR when using these augmented labels. For consistency, these augmentation strategies were not applied in other experiments.

E Supplementary Visualisations

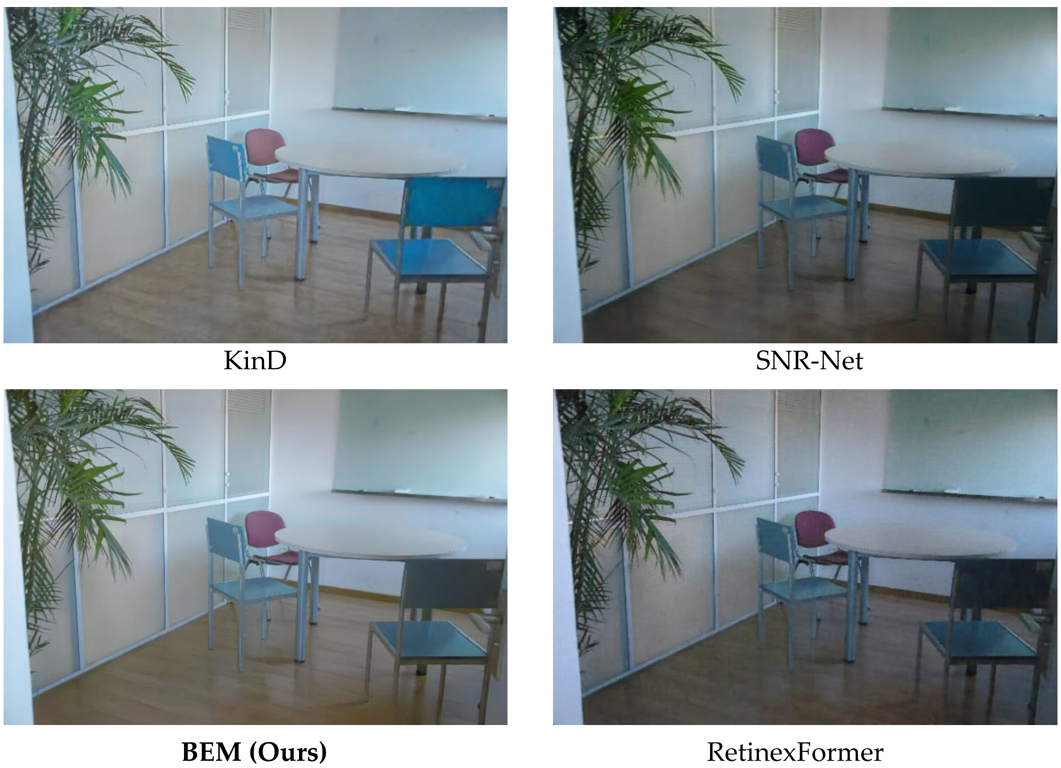


Figure 13: Visual comparisons with KinD, SNR-Net and RetinexFormer under images' original resolution. The sample is from the LOL-v2-real dataset.

HD Visualisation for LLIE. To facilitate a closer inspection of enhanced image details, we present high-resolution visual comparisons in Figure 13, where the predictions of state-of-the-art models are displayed at their original resolutions. The high-resolution visualisation reveals that previous state-of-the-art methods tend to exhibit varying degrees of noise artefacts in the enhanced results, significantly degrading perceptual quality. In contrast, our method effectively suppresses these noise artefacts, which are often introduced by low-light conditions. Furthermore, our approach achieves superior detail restoration, while other methods show signs of blurring and detail loss.

More Visualisations for UIE. In Figure 14, we present additional visual comparisons on the U45 and UCCS datasets, demonstrating that our method consistently outperforms PUGAN and PUIE-MP in enhancing various underwater scenes.

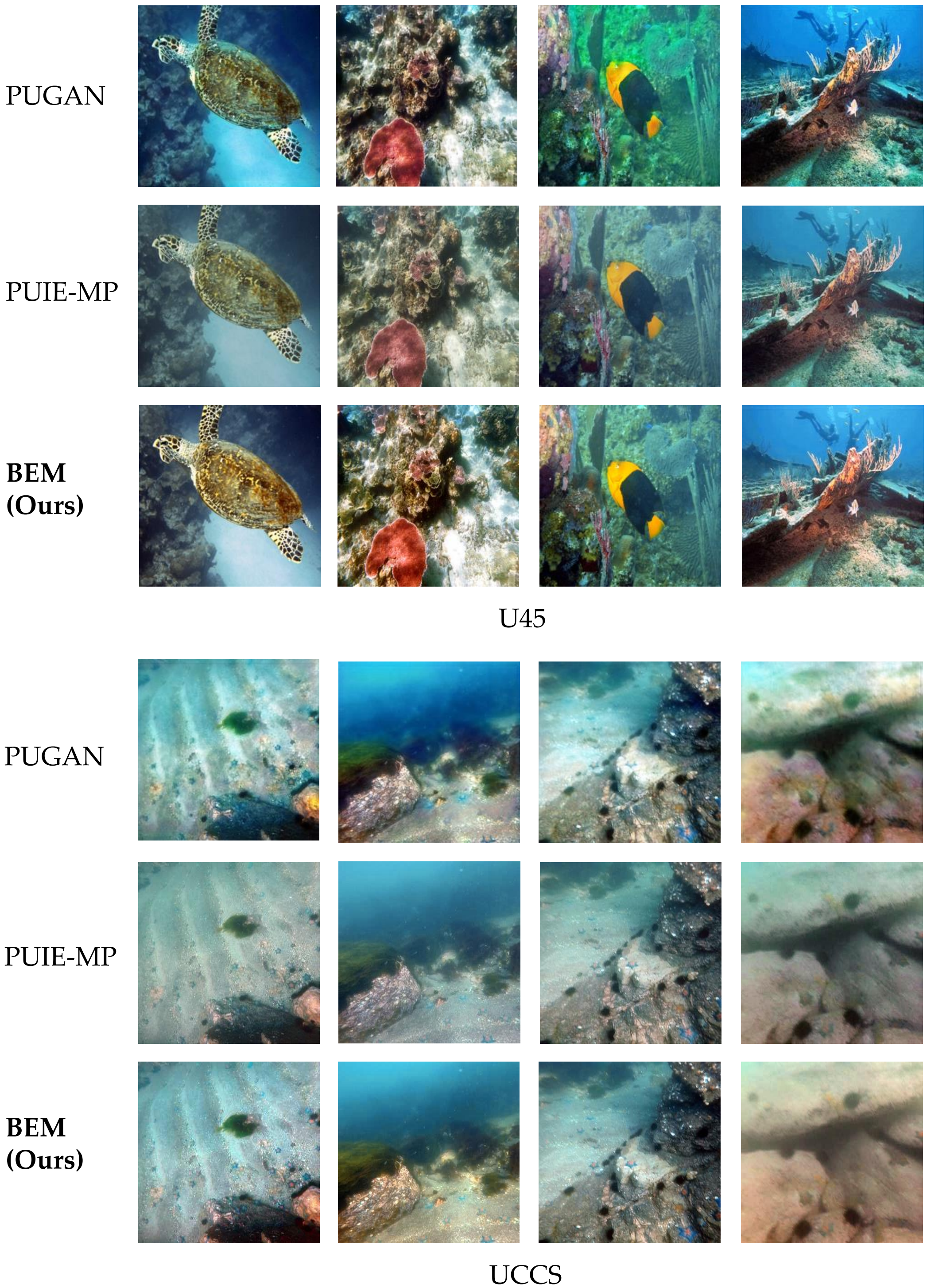


Figure 14: Visual comparisons with PUGAN and PUIE-MP on the U45 and UCCS test sets.

F Motivation of the Two-Stage Framework

To demonstrate the advantages and necessity of our two-stage BNN-DNN framework, we analyze its performance by comparing it with five other frameworks, as shown in Figure 15. The corresponding results on UIEB and LOL-v1 are presented in Table 8.

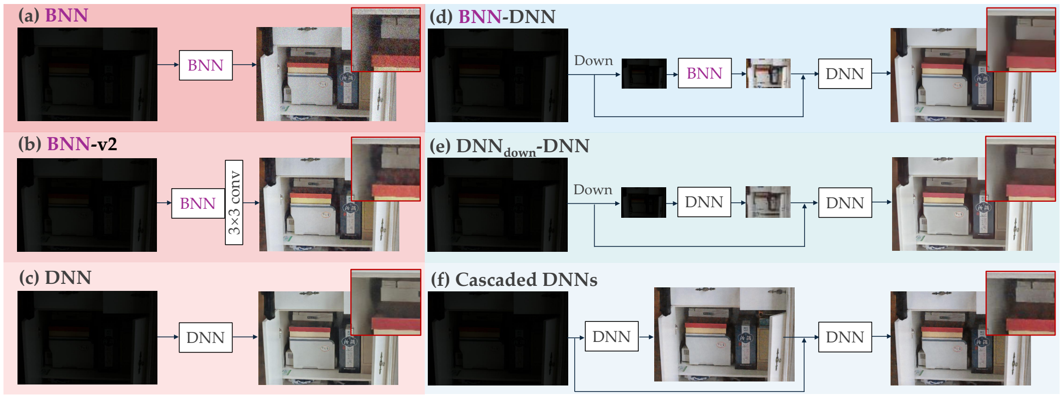


Figure 15: Illustration of six framework variants, including three one-stage models (a, b, and c) on the left and three two-stage models (d, e, and f) on the right. The arrows indicate the inference process, with each framework demonstrating different architectural designs. The square box labelled "Linear" in (e) denotes that the final projection layer is a deterministic linear layer. In (d) and (e), the first stage and second stage are training independently, while the two stages of Cascaded DNNs (f) are training together. Enlarged views highlight key regions for better comparison.

F.1 Limitations of One-Stage BNN

In high-dimensional image data, BNN introduces uncertainty in the prediction of each pixel. As shown in Figure 15 (a-b) and Figure 16, this pixel-level uncertainty manifests as noise in the output image, which negatively impacts both visual perception and certain image quality metrics. Nevertheless, the one-stage BNN models yet provide better results than pure DNN-based models. Visually, for example, by comparing the enlarged views of Figure 15 (a) and Figure 15 (c), we can observe that the BNN model is capable of recovering the red colour of the top surface of the box, while the DNN fails to do so. To cancel the noise in the enchanted image, we attempt to strengthen the spatial relations between adjacent pixels by retaining the BNN's output layer as a deterministic 3×3 convolutional layer as shown in Figure 15 (b). However, the denoising effect of this simple method is not satisfactory, and because the deterministic layer is introduced in the end-to-end training, the diversity of the model output is reduced.

F.2 Resort to the Two-Stage BNN-DNN Framework

In BNN-v2 (b), by removing the uncertainty in the weights of the final convolutional layer, specifically by eliminating the random noise term $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ in Eq. 6, we were able to significantly reduce the noise frequency. This leads us to hypothesize that the strong Gaussian-like noise observed in the output of BNN is primarily caused by the noise term ϵ in each Bayesian layer. Therefore, to eliminate the noise in the output, it becomes necessary to replace the Bayesian layers near the output of the BNN model with deterministic layers. However, this approach is not straightforward, as making the layers near the output deterministic inherently makes the entire output deterministic, effectively neutralizing the uncertainty provided by the BNN. To address this, we propose splitting the model into a BNN part and a DNN part, and training them separately. This forms the basis of our two-stage BNN-DNN framework.

Table 8: Comparisons of various one-stage and two-stage frameworks. For two-stage frameworks, the second column specifies whether 16× downsampling is applied to the input in the first stage.

Framework	Downscale (Stage-I)	UIEB-R90		LOL-v1	
		PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
(a) BNN	N/A	21.72	0.885	22.74	0.818
(b) BNN-v2	N/A	23.71	0.899	24.78	0.852
(c) DNN	N/A	20.83	0.864	23.76	0.842
(d) **BNN-DNN**	✓	**25.62**	**0.940**	**26.83**	**0.877**
(e) DNN_{down}-DNN	✓	20.68	0.812	22.85	0.823
(f) Cascaded DNNs	✗	20.95	0.873	23.98	0.827
(g) BNN-DNN	✗	17.78	0.689	19.26	0.798

To demonstrate the benefits of our separate training scheme, we compare it with cascaded DNNs (c), where both stages are trained jointly. As shown in Table 8, the two-stage separate training scheme outperforms the conventional cascaded DNNs. Meanwhile, we conduct an ablation study on the BNN component of the two-stage framework (d). Specifically, we replace the BNN part with a DNN of equivalent size, resulting in the DNNdown framework (e). By comparing the performance of both frameworks across different datasets, as shown in Table 8 , we observe that the BNN-DNN framework outperforms DNNdown. This result verifies that the primary performance improvement of the two-stage BNN-DNN framework is attributed to the BNN.

F.3 Importance of Input Downsampling for Stage-I

The input dimensionality reduction in the first stage of our BNN-DNN framework is crucial for the successful training of the second-stage model. This is because the two stages are trained independently, and during the training of the second stage, the predictions from the first stage are replaced with ground-truth (GT) information. Without dimensionality reduction, the training of the second stage becomes invalid, as it would merely result in learning an identity mapping, as evidenced by the result shown in the last row in Table 8. Furthermore, the BNN in the first stage is trained on downsampled, low-resolution images. We found that BNNs are more effective when dealing with these lower-dimensional data. In Table 9, we compare the performance of the BNN trained on 16× downsampled image datasets with its performance on the original resolution datasets. Our results show that the BNN achieves more accurate predictions when processing lower-resolution images compared to high-resolution images. In contrast, the DNN shows no obvious difference in predictive performance between low-resolution and original-resolution images.

Table 9: Comparing the performance of one-stage BNN on 16× downsampled image data of LOL-v1 and that of the original resolution LOL-v1.

Model	dataset	PSNR ↑
BNN_{down}	16× down LOL-v1	25.43
BNN	LOL-v1	22.74
DNN_{down}	16× down LOL-v1	22.25
DNN	LOL-v1	23.76

In Figure 16, we compare the enhanced outputs of the one-stage and two-stage models. The one-stage model's output exhibits noticeable noise due to the per-pixel uncertainty predictions of the BNN, whereas the two-stage model produces a noise-free output.


One-Stage **Two-Stage**

Figure 16: A visual comparison of enhanced images produced by the one-stage BNN-v2 (left) and two-stage BNN-DNN models (right).

G ANALYSIS OF PREDICTIVE UNCERTAINTY

In this section, we statistical analyse of the diversity in predictions generated by BEM. Table 10 presents the predictive uncertainty statistics collected from the LOL-v1 dataset. A larger standard deviation indicates higher uncertainty, suggesting that the BEM produces more diverse predictions and better captures the one-to-many mapping nature of the task. The maximum values approximate the upper bound of the BEM's predictive quality, while the minimum values approximate its lower bound.

Table 10: Statistic data on predictive uncertainty on LOL-v1. CLIP (Brightness) indicate the CLIP feature similarity using text prompt "`Bright photo`". Likewise, CLIP (Quality) use prompt "`Good photo`".

Metric	Maximum	Mean	Median	Minimum	Standard deviation
PSNR	26.89	22.87	22.97	17.90	1.911
SSIM	0.876	0.855	0.856	0.819	0.013
CLIP-IQA (Brightness) $\times 100$	93.62	89.63	89.71	84.20	1.689
CLIP-IQA (Quality) $\times 100$	64.34	59.13	59.08	54.22	1.825
CLIP-IQA (Noisiness) $\times 100$	36.17	30.06	30.02	25.08	1.942
Negative NIQE	- 4.647	-4.808	- 4.806	-4.971	0.059

As shown in Table 10, the minimum CLIP-IQA values in the LOL dataset are significantly smaller than the maximum values, potentially reflecting the presence of low-quality GT images in the dataset. We hypothesise that these poor-quality GT images significantly impact the performance of deterministic neural networks. However, due to BEM's uncertainty modelling, such low-quality GT images primarily affect the lower bound of BEM's predictive quality, minimising their overall influence on performance.

In Figure 17, we randomly selected an input image from the heterogeneous dataset LSRW (Hai et al., 2023) to analyse the distribution of its prediction results. We observe that, for each metric, although many predictions fall within the central range, they are not overly concentrated. This demonstrates the diversity of the model's predictions.

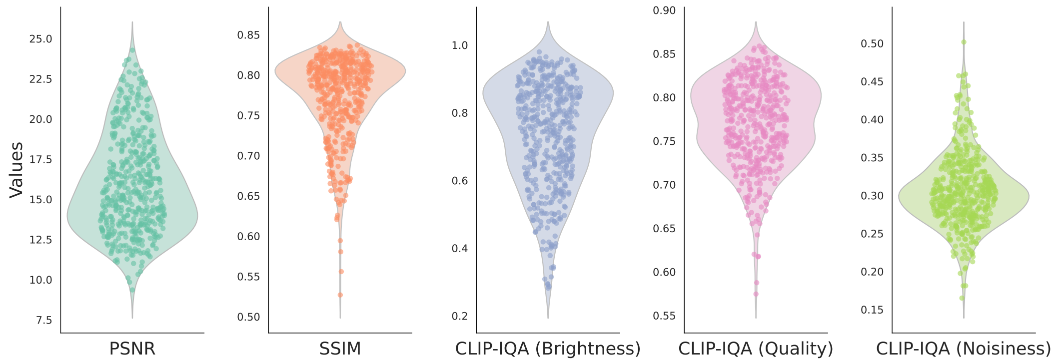


Figure 17: Distribution of 500 random predictions generated by the BEM model for a single low-light image across different evaluation metrics, including PSNR, SSIM, and three CLIP-IQA metrics (“Brightness”, “Quality”, “Noisiness”). Each violin plot visualises the density and range of predictions.

From the uncertainty map (e) in Figure 18, we observe a structured distribution of uncertainty, where regions expected to be in shadow exhibit lower uncertainty, while illuminated areas tend to have higher uncertainty.

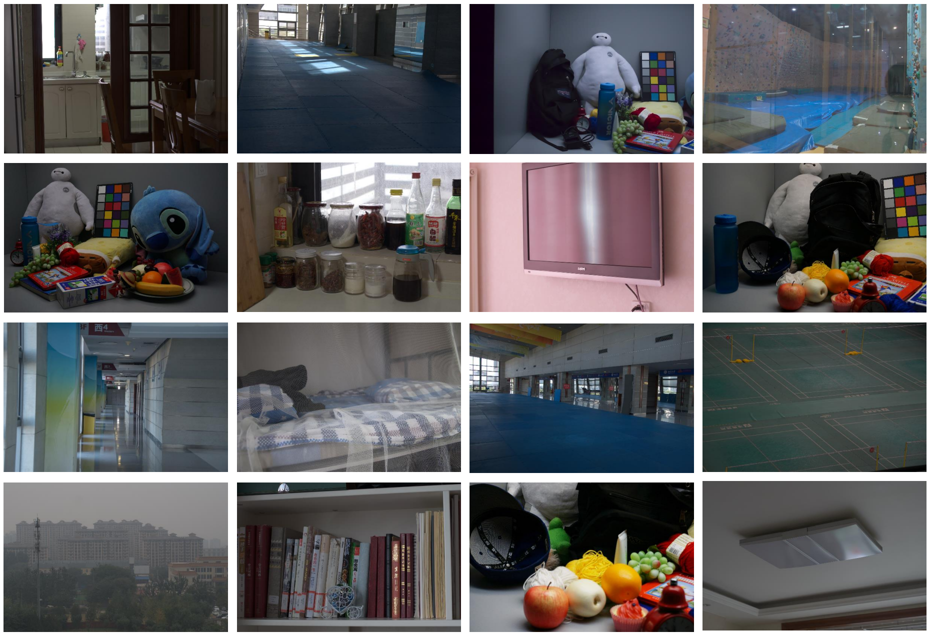
(a) Input (b) GT (c) Max. (PSNR) (d) Min. (PSNR) (e) Uncertainty

Figure 18: Visualisation of BEM outputs showing the input image (a), ground truth (b), the prediction with the highest PSNR (c), the prediction with the lowest PSNR (d), and the uncertainty map (e). The uncertainty is computed as the pixel-wise standard deviation across 500 predicted images.

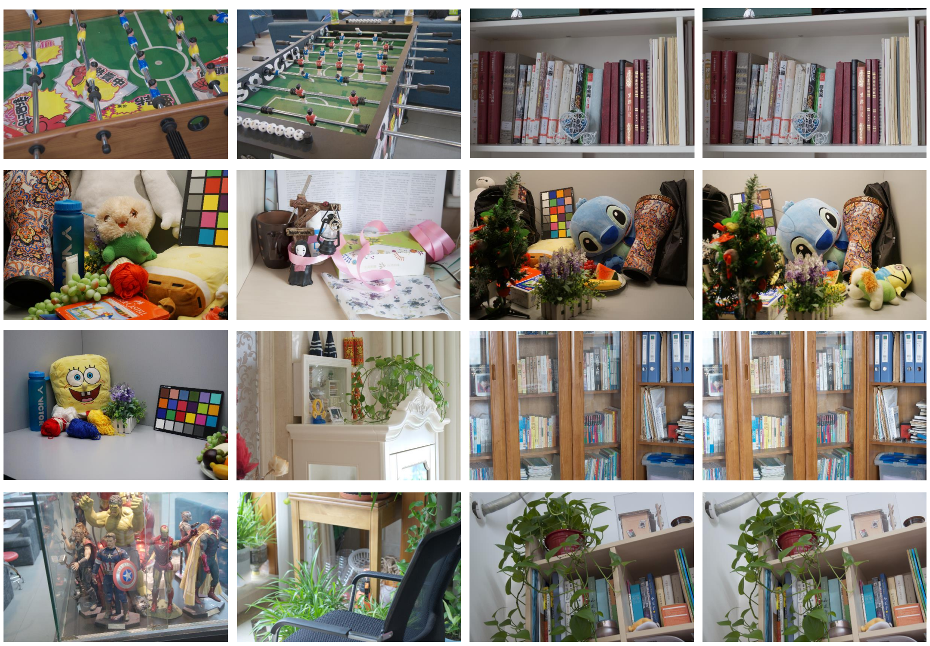
To investigate how the predictive uncertainty and quality of BEM are influenced by the overall GT quality in the training data, we conduct the following experiments as detailed in Appendices G.1 and G.2.

G.1 Step One: Identify low-quality GT images in Training Data

To separate training data with low-quality GT images from the dataset, we initially employed CLIP-IQA (Wang et al., 2023) with text prompts (“Brightness”, “Noisiness”, “Qualit”) to filter out images with low brightness, high noise levels, and poor quality. This automated process was followed by manual refinement to identify and separate poor-quality GT images. Examples of low-quality GT images from the LOL and UIEB training sets are shown in Figure 19 and Figure 20, alongside high-quality GT images for comparison. While the algorithmic filtering reduced excessive subjectivity, the manual refinement process may still introduce some subjective bias. Therefore, the separation results should be treated as indicative rather than definitive.

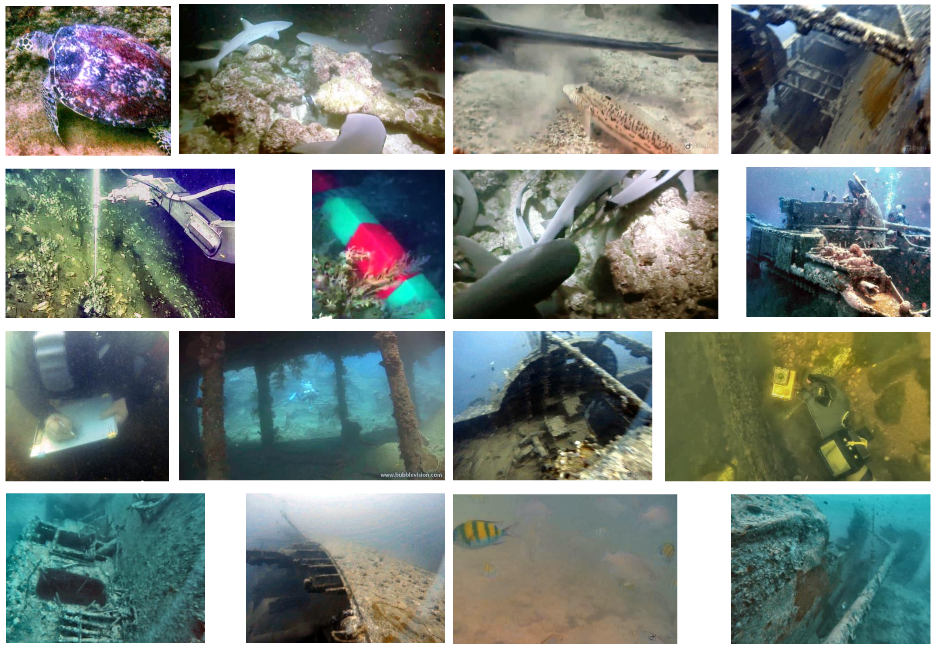


Low-Quality GT Images

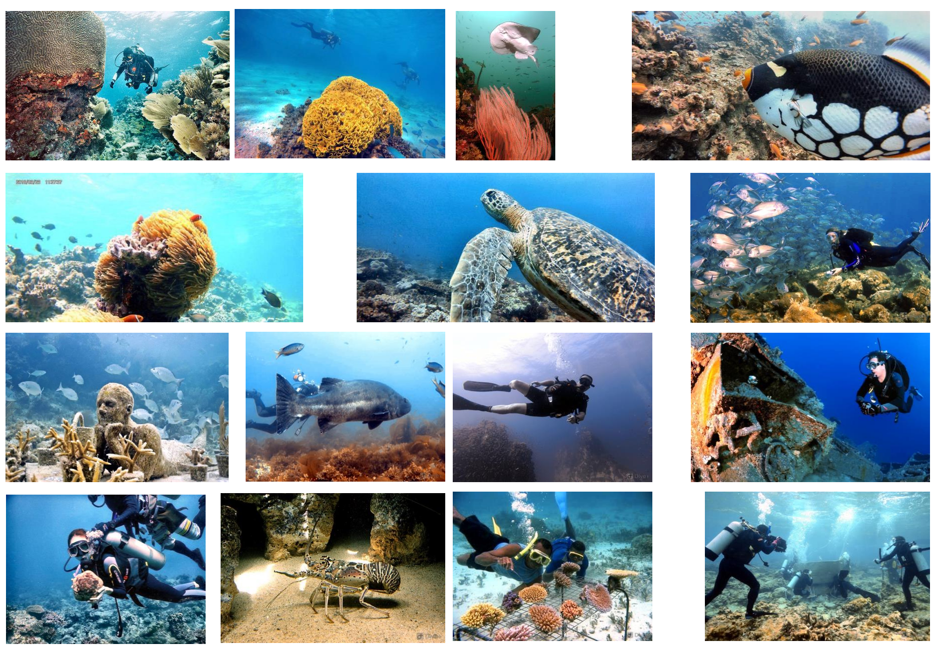


High-Quality GT Images

Figure 19: Examples of low-quality and high-quality GT images from the LOL training set. The categorisation may be influenced by subjective biases in assessing visual clarity, lighting, and overall image quality.



Low-Quality GT Images



High-Quality GT Images

Figure 20: Examples of low-quality and high-quality GT images from the UIEB training set. The categorisation may be influenced by subjective biases in assessing visual clarity, lighting, and overall image quality.

G.2 Step Two: Impact of Training Data Quality on Predictive Performance

When the dataset contains low-quality ground-truth images, BEM generates a distribution of predictive quality, producing both high-quality and low-quality outputs. The probability of generating high-quality outputs is influenced by the proportion of high-quality ground-truth images in the training data.

Specifically, as the proportion of high-quality ground-truth images increases, the probability of sampling high-quality outputs during inference also rises. Consequently, fewer sampling iterations are required to obtain satisfactory enhancement results. Conversely, when the proportion of high-quality ground-truth images is low, more sampling iterations are needed.

To examine whether the proportion of high-quality ground-truth (GT) images in the training data affects the likelihood of generating high-quality outputs, we pose the question: Does increasing the share of high-quality images in the training set improve the probability of producing high-quality results?

To test this hypothesis, we conducted the following experiment: First, using the sample separation method described in Sec. G.1, we identified and labelled low-quality GT images in the training dataset. Next, while keeping the total size of the training dataset constant, we systematically replaced low-quality GT images in the LOL-v1 training set with high-quality GT images from the LOL-v2-real dataset. This allowed us to control the proportion of high-quality images in the training data, denoted as τ.

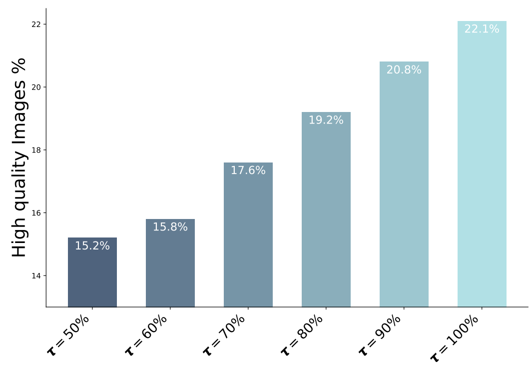


Figure 21: Impact of training data quality on BEM. The x-axis represents the proportion of high-quality images in the training dataset (τ), while the y-axis shows the percentage of high-quality predictions obtained after $K = 100$ sampling times on the test set. Higher proportions of high-quality training data lead to a greater likelihood of generating high-quality predictions. A prediction is classified as high-quality if its CLIP (Quality) score exceeds 0.8.

The results, shown in Figure 21, demonstrate a clear trend: as the proportion of low-quality GT images decreases, the likelihood of generating high-quality outputs increases consistently. When the training dataset consists entirely of high-quality GT images ($\tau = 100\%$), BEM achieves significant efficiency, producing a satisfactory enhanced output approximately once every five sampling iterations on average. This highlights the direct relationship between training data quality and the predictive performance of BEM. Nonetheless, the true strength of BEM lies in its ability to generate high-quality enhanced images even when real-world data contains low-quality GT images, thanks to its uncertainty modelling capabilities. The trade-off, however, is the need for more sampling attempts.

H USE CLIP TO PICK OUT A HIGH-QUALITY ENHANCED IMAGE

As illustrated in Figure 22, the ground-truth images in the test set are low-quality. When evaluated using full-reference metrics such as MSE or PSNR, BEM produces outputs like image (b), which closely resemble the low-quality GT image. In contrast, when using CLIP-IQA as a no-reference metric, BEM generates outputs like image (a). Upon observation, image (a) demonstrates superior illumination and clarity compared to image (b) in Figure 22.

Figure 23 illustrates the outputs selected by BEM using the no-reference CLIP metric and the full-reference PSNR metric, alongside other unselected predictions. Notably, the results selected by both metrics are visually acceptable.

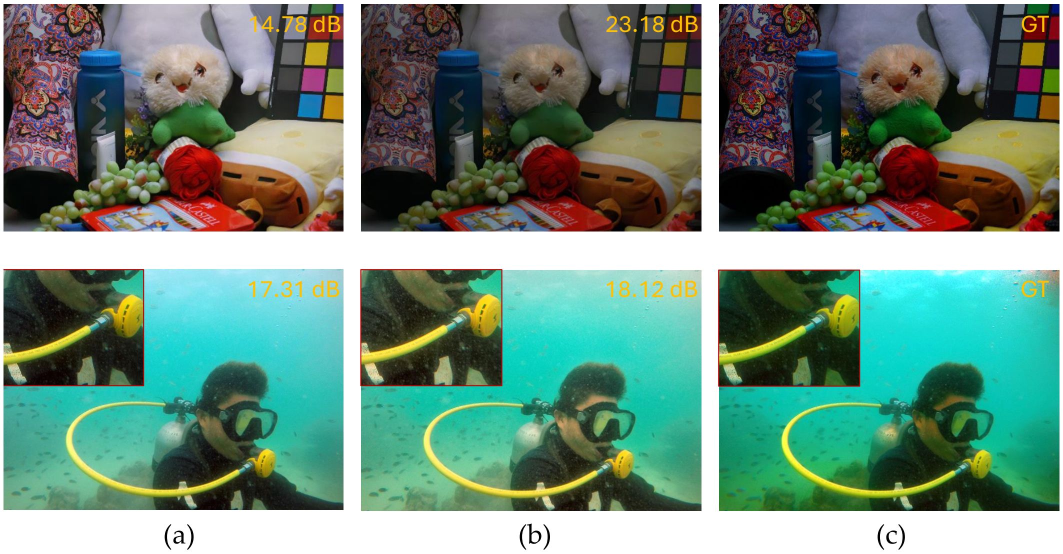


Figure 22: A superior enhancement does not necessarily align with the suboptimal ground truth. The left and middle images represent two plausible outputs from BEM, showcasing diverse enhancements. The left images are selected using the no-reference CLIP-IQA (Qualify) metric, while the middle images are chosen based on the full-reference PSNR metric.

Figure 23: Visualisation of BEM predictions. The pink box (□) highlights the output selected using the no-reference CLIP-IQA ("Brightness", "Noisiness", "Quality") metric, while the blue box (□) highlights the output selected using the full-reference PSNR metric. The input image is from the LSRW dataset (Hai et al., 2023).

In Table 11, we present the results obtained by instantiating the quality metric D in Algorithm 1 as CLIP-IQA with the text prompts "Natural", "Brightness", and "Warm". Notably, we intentionally avoided using "Quality" as the prompt for CLIP, as it tends to select the highest-quality images. Given that some GT images in the LOL-v1 dataset are of suboptimal quality, this choice could result in a decrease in full-reference metrics like PSNR.

I Additional Results on UIEB

Table 11: Additional quantitative results of BEM using CLIP-IQA (denoted as BEM_{CLIP}) on the LOL-v1 and v2 datasets. GT Mean is used to adjust the output brightness. The BEM model use full-reference quality metric is denoted as BEM_{full}.

Method	LOL-v1			LOL-v2-real			LOL-v2-syn		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
BEM	28.80	0.884	0.069	32.66	0.915	0.060	32.95	0.964	0.026
BEM_{CLIP}	28.43	0.882	0.071	30.01	0.910	0.076	31.51	0.961	0.030
$\text{BEM}_{\text{Determ.}}$	28.30	0.881	0.072	31.41	0.912	0.064	30.58	0.958	0.033

In Table 12, we provide additional results on the validation set of UIEB in terms of FID and LPIPS. The listed methods includes UIEC^2-Net (Wang et al., 2021), Water-Net (Li et al., 2019a), U-color (Li et al., 2021), U-shape (Peng et al., 2023), DM-water (Tang et al., 2023), PA-Diff and (Zhao et al., 2024b) WFI2-net (Zhao et al., 2024a).

Table 12: Results on UIEB in terms of FID and LPIPs.

Method	UIEC^2-Net	Water-Net	U-color	U-shape	DM-water	PA-Diff	WFI2-net	BEM (ours)
FID ↓	35.06	37.48	38.25	46.11	31.07	28.74	27.85	**26.11**
LPIPS ↓	0.2033	0.2116	0.2337	0.2264	0.1436	0.1328	0.1248	**0.1019**

(a) (b) (c)

Figure 24: (a) Input image; (b) input image after linear brightness adjustment; (c) output of the one-stage BNN. When the input photo is particularly dark, the read noise becomes more prominent after brightness adjustment, making its impact on the output more noticeable. This suggests that the one-stage BNN might amplify such noise unintentionally due to its inherent uncertainty, leading to less desirable output results.