Degradation-Robust Fusion: An Efficient Degradation-Aware Diffusion Framework for Multimodal Image Fusion in Arbitrary Degradation Scenarios

Supplementary Material

A. Diffusion Models

A.1. Denoising Diffusion Probabilistic Models

Denoising Diffusion Probabilistic Models (DDPM) are a class of generative models that rely on a forward process of gradually adding noise to data, followed by a reverse process to recover the original data. The key idea behind DDPM is to model the distribution of the data using a Markov chain, where each step in the chain progressively adds noise until the data is destroyed into pure noise. The reverse process then attempts to reverse this noise addition, thereby generating samples from the data distribution.

Forward Diffusion Process. In the forward process, noise is gradually added to the data, making it more random at each step. Formally, the forward process can be described as a sequence of noisy data points $\{x_0, x_1, \ldots, x_T\}$, where the data at each step x_T is obtained by adding Gaussian noise to the previous step. The forward process is typically defined by a variance schedule $\beta_1, \beta_2, \ldots, \beta_T$.

The forward process can be described as:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}), \tag{1}$$

where β_t controls the noise added at each step, and x_t is the noisy version of the data at time step t.

The distribution of the data at time t is then:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}), \tag{2}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$.

Reverse Process: The reverse process involves learning a model that can reverse the noise addition, transforming pure noise back into a sample from the data distribution. This reverse process is modeled by a neural network $\epsilon_0(x_t, t)$, which predicts the noise added at each time step.

The reverse process can be written as:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_\theta(t)^2\mathbf{I}), \tag{3}$$

where $\mu_\theta(x_t, t)$ and $\sigma_\theta(t)$ are the mean and variance predicted by the neural network. The model is trained to minimize the difference between the true noise and the predicted noise at each step.

Training Objective: The objective is to learn the reverse process by minimizing the evidence lower bound (ELBO). This is typically done by minimizing the variational bound on the negative log-likelihood, which leads to the following loss function:

$$L_{\text{DDPM}} = \mathbb{E}_{q(x_0,x_T)}\Bigg[\sum_{t=1}^{T}\mathbb{E}_{q(x_t|x_{t-1})}\Big[\|\epsilon_\theta(x_t, t) - \epsilon^*(x_t, t)\|^2\Big]\Bigg], \tag{4}$$

where $\epsilon^*(x_t, t)$ is the true noise added at each step.

A.2. Denoising Diffusion Implicit Models

Denoising Diffusion Implicit Models (DDIM) are a variant of DDPM that introduce a more efficient sampling process. While DDPM generates samples by following the reverse process step by step, DDIM allows for implicit sampling, meaning that fewer steps are required to generate samples without sacrificing sample quality. DDIM achieves this by changing the reverse diffusion process, allowing for a deterministic trajectory of the reverse process.

Reverse Process in DDIM: The key difference in DDIM is the deterministic nature of the reverse process. Instead of adding Gaussian noise at each step, DDIM defines a reverse process with a fixed noise schedule, leading to fewer steps needed to generate high-quality samples.

The reverse process in DDIM can be defined as:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_\theta(t)^2\mathbf{I}), \tag{5}$$

where the mean $\mu_\theta(x_t, t)$ is determined by the model and the noise variance $\sigma_\theta(t)$ is fixed in DDIM. The reverse process is defined in such a way that the trajectory of the reverse diffusion is deterministic, leading to a more efficient sampling procedure. DDIM leverages the forward noising formula in DDPM and the reparameterization technique to transform the original x_t into a deterministic mapping form of x_0, and ultimately derives the following iterative for-

mula:

$$x_{t-1} = \sqrt{\alpha_{t-1}} \left(x_t - \sqrt{1-\alpha_t} \epsilon_\theta^{(t)}(x_t) \right) \frac{1}{\sqrt{\alpha_t}} + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \epsilon_\theta^{(t)}(x_t) + \sigma_t \epsilon_t \tag{6}$$

This iterative formula is typically used in the context of denoising diffusion models (e.g., DDIM). It describes the process of updating the latent variable x_{t-1} based on the current latent x_t, the deterministic mapping x_0, and a noise term. The terms α_t, α_{t-1}, and θ are parameters related to the denoising process, and the formula ensures the iterative update is consistent with the model's reparameterization and forward noising mechanisms.

Algorithm 1 Efficient Degradation-Aware Diffusion Framework for Image Fusion

Input: Degraded source images $\mathbf{y}_1$, $\mathbf{y}_2$, degradation operators $\mathbf{A}_1$, $\mathbf{A}_2$, maximum iteration steps T.
Output: $\mathbf{X}_f$

Construct the joint observation vector using Eq. (12).
Initialize the input $\hat{\mathbf{x}}_T$ using the weighted average of the source images.
for $t = T$ **down to** 1 **do**
  Predict noise $\epsilon_\theta(\hat{\mathbf{x}}_t, t)$ and fusion weight $\mathbf{W_1}$ using multi-task U-Net θ.
  Calculate complementary fusion weight: $\mathbf{W}_2 = 1 - \mathbf{W}_1$.
  Construct the joint degradation matrix $\hat{\mathbf{A}}$ via Eq. (13).
  Implicitly compute the pseudoinverse $\hat{\mathbf{A}}^\dagger$ via Eq. (15).
  Compute the unconstrained denoised estimate $\hat{\mathbf{x}}_{0|t}$ via Eq. (16).
  Perform joint degradation-aware correction to obtain $\overline{\mathbf{x}}_{0|t}$ via Eq. (17).
  Update the latent state to $\hat{\mathbf{x}}_{t-1}$ via Eq. (18).
end for
Extract the final fused image component $\mathbf{X}_f$ from $\hat{\mathbf{x}}_0$.
return $\mathbf{X}_f$

B. Comprehensive Analysis and Refinement of Joint Constraint Correction

In image fusion tasks, the core idea of the joint constraint correction mechanism is to introduce multiple constraints, ensuring that the fusion image not only satisfies the observation consistency of each source image but also maintains overall fusion consistency. In this mechanism, the degradation process and fusion process are coupled together, and linear constraints ensure that the relationship between each source image and the fused image is effectively constrained.

Joint Variables and Constraint Design: Let the joint variable be $\boldsymbol{x} = [x_1, x_2, x_f]$, where x_1 and x_2 are the two source images, and x_f is the fused image. Three types of constraints are introduced to describe the relationship between the source images and the fused image:

Data Consistency (Two-way) Constraint:

$$\mathbf{y}_1 = A_1 \mathbf{x}_1 + \mathbf{n}_1, \tag{7}$$

$$\mathbf{y}_2 = A_2 \mathbf{x}_2 + \mathbf{n}_2, \tag{8}$$

where y_1 and y_2 are the observations of the two source images, A_1 and A_2 are the degradation operators, and n_1 and n_2 are the noise terms.

Fusion Consistency (Linear) Constraint:

$$\mathbf{x}_f = W_1 \mathbf{x}_1 + W_2 \mathbf{x}_2, \tag{9}$$

where W_1 and W_2 are the linear fusion operators, which can be learned during training. The fusion operators describe the linear relationship between the source images and the fused image.

These constraints can be combined into an overall linear equation system, using the joint degradation matrix A and the joint observation y. Specifically, the degradation matrices A_1 and A_2 can be designed according to the actual task, while the fusion operators W_1 and W_2 can be learned through training, typically frozen during the internal update at each step. The joint degradation matrix can be written in the following form:

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 & 0 & 0 \\ 0 & \mathbf{A}_2 & 0 \\ -\mathbf{W}_1 & -\mathbf{W}_2 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \mathbf{X}_f \end{bmatrix}. \tag{10}$$

Pseudo-inverse and Correction Mechanism: For the above linear constraints, the goal of the correction process is to find a new solution x^* that satisfies the constraints and minimizes the Euclidean distance to the initial solution x_0. Geometrically, this problem is equivalent to orthogonally projecting the initial solution x_0 onto the hyperplane defined by the linear equation $Ax = y$, where A is the joint matrix. The correction amount can be obtained by solving the following linear equation system:

$$Ax^* = y, \tag{11}$$

Or write it in an optimized form:

$$\mathbf{x}^* = \arg\min_z \|z - x_0|_t\|^2 \quad \text{s.t. } Az = y. \tag{12}$$

In numerical computation, we typically do not directly calculate the pseudo-inverse $A^\dagger$ because it may be expensive and unstable. Instead, we compute the correction amount:

$$\mathbf{x}^* = \mathbf{x}_0 - \Delta\mathbf{x}, \tag{13}$$

where $\Delta \mathbf{x}$ is the correction amount, representing the shift from the initial solution to the optimal solution that satisfies the constraints. Specifically, the solution to the above equation is:

$$x^\star = x_0 - A^\dagger(Ax_0 - y). \tag{14}$$

The correction amount is typically solved using Conjugate Gradient (CG), which is an efficient iterative method for large-scale problems (e.g., high-resolution images).

Joint modeling and correction offer significant benefits by ensuring consistency and coherence across multiple data sources. By simultaneously considering various constraints, this approach maintains the interdependencies between source images and the fused image, leading to more accurate and realistic results. It efficiently integrates available information, handles complex degradations, and improves computational efficiency. Moreover, joint correction minimizes errors by ensuring that the solution satisfies all constraints, even in the presence of noisy or incomplete data. This makes joint modeling particularly effective in unsupervised learning tasks and large-scale applications. However, directly using the Conjugate Gradient method for explicit computation is not feasible in practice due to the risk of memory explosion and high computational costs, as the method requires multiple iterations. Therefore, we employ an alternative approach to obtain the pseudo-inverse based on the solution of the equation. The details of this method can be found in Section 3 of the main text.

C. Degradation Definition

Here's a detailed description of the three operations—noise addition, blurring, and super-resolution—and their corresponding A (degradation matrices) and $A^\dagger$ (pseudo-inverse) in the context of the proposed model.

Noise Addition: Noise addition is a common degradation process where random noise is introduced to the original image. Mathematically, this operation can be expressed as:

$$y = Ax + n. \tag{15}$$

The pseudo-inverse of the degradation matrix A in the case of noise addition is simply the identity matrix. In the diffusion process, to reduce the impact of noise, we apply an additional intensity control coefficient to the noise-containing correction term. This approach is inspired by the DDNM.

Blurring: In image restoration tasks, the blur operator A typically represents a linear degradation process, where an image x is convolved with a blurring kernel k to produce a degraded image y, i.e., $y = Ax = k * x$, where * denotes the convolution operation. This degradation process can be seen as a linear system where the blurring kernel k acts as a filter that removes or distorts certain image details.

To restore the original image x from the blurred observation y, we need to compute the pseudo-inverse Ap of the blur operator. In the frequency domain, Wiener convolution provides an optimal solution to this problem by minimizing the mean square error between the true and estimated images. The Wiener filter in the frequency domain is given by:

$$H(\omega) = \frac{H^*(\omega)}{|H(\omega)|^2 + \gamma}, \tag{16}$$

Where $H(\omega)$ is the Fourier transform of the blurring kernel k, γ is a regularization term that accounts for noise in the observation, and $H^*(\omega)$ is the complex conjugate of $H(\omega)$. This filter effectively acts as a frequency-domain approximation of the pseudo-inverse, restoring the image by compensating for the blurring and noise.

The Wiener convolution approach is ideal for this problem for several reasons: 1) Linear Degradation Model: The blur operator is linear, meaning the relationship between the observed and true image can be captured using a linear filter, making Wiener convolution a suitable choice for solving the inverse problem. 2) Frequency Domain Efficiency: By working in the frequency domain, the Wiener filter takes advantage of the fast Fourier transform (FFT), significantly speeding up the computation of the pseudo-inverse. 3) Noise Suppression: The regularization term γ in the Wiener filter helps mitigate the amplification of noise, ensuring that the restored image is not overly influenced by noise in the observation. Thus, using Wiener convolution to compute Ap provides an effective and computationally efficient method to reverse the blurring process and recover the original image, especially in the presence of noise.

Low Resolution: The degradation operator A represents the downsampling operation applied to the original image to simulate a lower-resolution observation. In this case, the degradation process is modeled by an adaptive average pooling operation, which reduces the image resolution by a factor determined by the scaling parameter. The pseudo-inverse operator Ap corresponds to the upsampling operation, where the low-resolution image is transformed back to a higher-resolution image. This operation is achieved by the PatchUpsample function, which increases the spatial dimensions of the image. The PatchUpsample function upsamples the low-resolution image x by a factor of scale. This operator restores the image to its higher resolution by expanding the spatial dimensions (height and width) of the input image, effectively reversing the downsampling process. It does this by distributing the pixel values of the low-resolution image into the larger output grid.

D. Ablation Experiments

In the main body of the paper, we did not present the specific results and detailed metrics of the ablation experiments. These results will be provided in this supplementary material. The analysis will focus on the performance of the

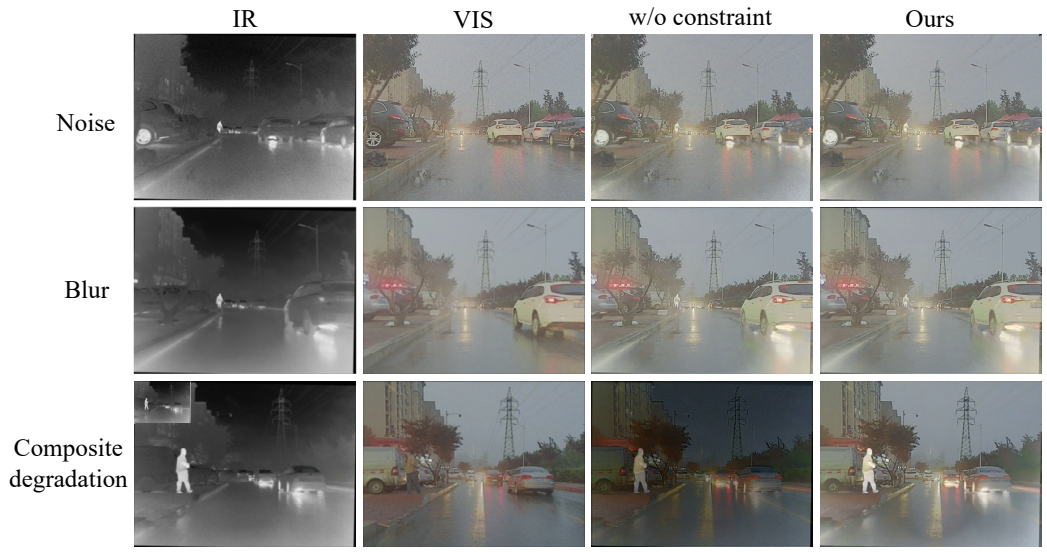

Figure 1. Results with and without the joint constraint correction mechanism under different degradation scenarios on M3FD dataset.

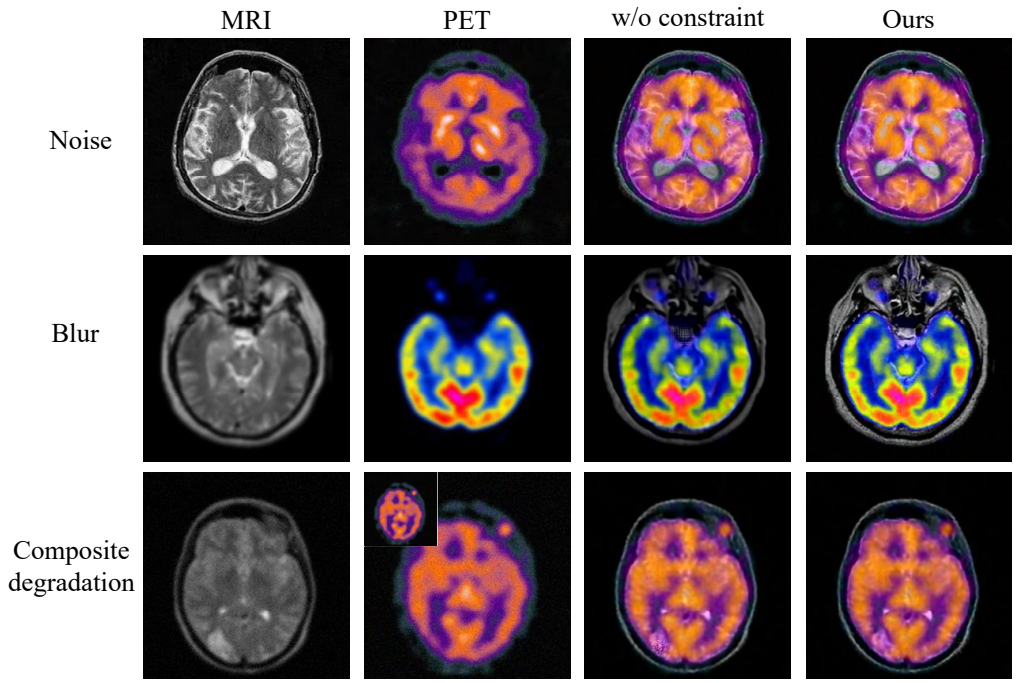

Figure 2. Results with and without the joint constraint correction mechanism under different degradation scenarios on PET-MRI dataset.

fusion results in three different tasks: denoising, deblurring, and compound degradation. The results on the M3FD and Harvard Medical datasets are shown in Fig 1 and Fig 2, respectively.

From these figures, it is evident that removing the proposed joint constraint correction mechanism leads to a noticeable degradation in the fusion results. Specifically, the images become noisier, with less distinct details, and more edge artifacts appear. This effect is particularly pronounced in the M3FD dataset, which aligns with the intended degradation scenarios. The results suggest that, in more complex degradation conditions, the proposed correction mechanism plays a crucial role in improving the final fusion accuracy. In the denoising task, for example, the absence of the joint constraints causes the fused image to retain more noise, leading to poor preservation of structural details. Simi-

Table 1. Ablation results for M3FD and PET-MRI datasets under different degradation scenarios. The best values for each metric are highlighted in light gray.

	M3FD						PET-MRI					
	Noise		Blur		Composite degradation		Noise		Blur		Composite degradation	
Metrics	Ours	w/o constraint	Ours	w/o constraint	Ours	w/o constraint	Ours	w/o constraint	Ours	w/o constraint	Ours	w/o constraint
Q_{MI}	**0.3505**	0.3322	**0.4477**	0.4140	**0.3732**	0.2322	0.6171	**0.6185**	**0.6469**	0.6318	0.6019	**0.6116**
Q_{NCIE}	**0.8052**	0.8050	**0.8068**	0.8062	**0.8055**	0.8038	**0.8078**	0.8077	**0.8077**	0.8071	0.8073	**0.8074**
$Q^{AB/F}$	**0.4083**	0.3759	**0.3698**	0.3233	**0.2199**	0.1054	**0.4258**	0.4092	**0.3855**	0.2019	**0.2956**	0.2544
Q_P	**0.1825**	0.1454	**0.1671**	0.1409	**0.0755**	0.0449	**0.2436**	0.2336	**0.2016**	0.1199	**0.1258**	0.1196
Q_{CB}	0.4206	**0.4386**	**0.4567**	0.4445	0.3790	**0.4356**	**0.4595**	0.4577	**0.6281**	0.5909	**0.4873**	0.4788
Q_W	**0.7810**	0.7465	**0.7233**	0.6907	**0.6237**	0.4135	**0.7892**	0.7583	**0.7885**	0.5232	**0.7344**	0.6825

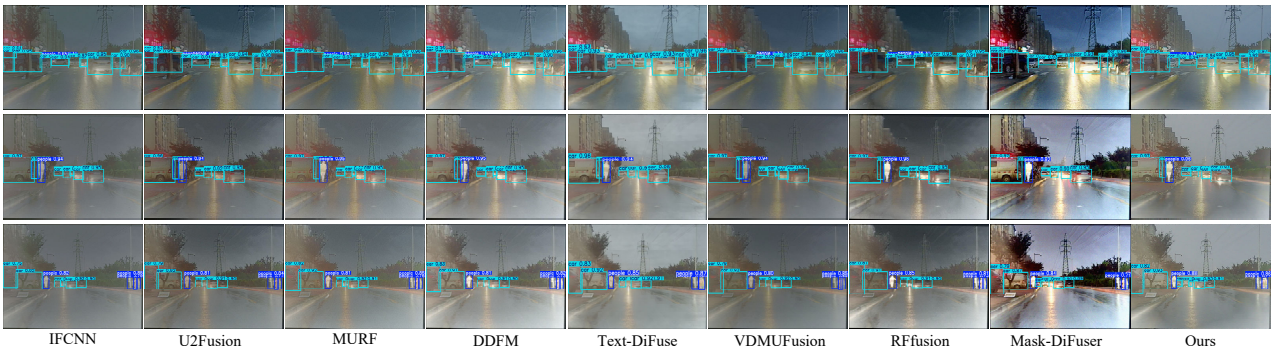

Figure 3. Qualitative detection results based on fusion images generated by different fusion methods.

Table 2. Detection performance comparison on M3FD dataset.

Method	Precision	Recall	mAP@0.5
IFCNN	0.9396	0.7933	0.8906
U2Fusion	0.9273	0.7605	0.8700
MURF	0.9510	0.7942	0.8865
DDFM	0.9620	0.7062	0.8509
Text-Difuse	0.9723	0.6546	0.8172
VDMUFusion	0.9690	0.7042	0.8499
RFfusion	0.9542	0.7504	0.8705
Mask-DiFuser	0.8769	0.6449	0.7891
Ours	**0.9750**	**0.8005**	**0.9108**

larly, in the deblurring task, without the proposed mechanism, the image sharpness significantly decreases, and blurring artifacts become more prominent. This trend is consistent across both datasets, indicating that the joint constraint correction mechanism is particularly effective in handling more complex degradation scenarios, where traditional fusion methods struggle to provide accurate reconstructions. Especially in the compound degradation scenarios of the M3FD dataset, it is evident that the more complex the scene, the more significant the improvement in fusion accuracy brought by the constraint correction mechanism.

The objective metrics, summarized in Table 1, present the results of the ablation study on the M3FD and PET-MRI datasets under different degradation scenarios (Noise, Blur, and Composite). The results show that our method outperforms the baseline (w/o constraint) across most metrics. Specifically, the proposed method achieves higher values for key metrics such as Q_{MI}, Q_{NCIE}, and Q_P, indicating better preservation of image details, structural integrity, and perceptual quality. These improvements are especially noticeable in more complex scenarios, such as composite degradation, where our method effectively preserves higher image quality and reduces degradation artifacts. Overall, the results highlight the significant benefits of the joint constraint correction mechanism in enhancing fusion performance across various degradation conditions.

E. Performance on High-level Vision Task

Image fusion is an effective form of image enhancement, whose ultimate goal is to facilitate subsequent high-level

vision tasks such as object detection in video surveillance and lesion segmentation in clinical diagnosis. Better fusion quality should naturally translate into better performance on downstream vision tasks. To verify the practical utility of the proposed model, we evaluate its detection performance under the most adverse degradation scenarios by conducting object detection experiments on the M3FD dataset using detection results obtained from different fusion methods.

As reported in Table 2, different fusion methods lead to clearly different detection performance on the M3FD dataset. Our method achieves the best results on all three metrics, with a precision of 0.9750, recall of 0.8005, and mAP@0.5 of 0.9108. Compared with the strongest baseline IFCNN, our approach improves mAP@0.5 by about 2.0 percentage points (from 0.8906 to 0.9108) while slightly increasing both precision and recall. Several competing methods, such as DDFM, Text-DiFuse and VDMUFusion, obtain relatively high precision (around 0.96–0.97) but suffer from noticeably lower recall (below 0.71) and mAP@0.5 (below 0.86), indicating that they tend to miss more targets. In contrast, our fusion model provides a more favorable balance between precision and recall, leading to the overall highest detection accuracy and confirming its effectiveness for downstream high-level vision tasks.

The visual results of the detection task are shown in Fig. 3. For the comparison methods such as IFCNN and U2Fusion, the pedestrian regions in the fused images are relatively dark, leading to suboptimal detection performance, while other methods also suffer from noticeable distortions in different areas, indicating limited fusion accuracy. In contrast, the results of our method are superior to all competitors both in terms of degradation removal and fusion quality. Most detected regions in our fused images achieve higher confidence scores than those obtained by all comparison methods, which further demonstrates the potential of the proposed approach for practical applications.

F. Analysis of the Parameter T

To investigate the impact of the iteration number T on the final fusion performance, we conduct an ablation study with T ranging from 1 to 5, as reported in Table 3. It can be observed that when T increases from 1 to 3, the objective metrics exhibit a continuous upward trend, indicating that the iterative mechanism effectively refines and enhances the fusion quality. The performance reaches its peak at $T = 3$, where our method achieves the best results on most metrics, including Q_{MI}, $Q^{AB/F}$, Q_P, Q_{CB}, and Q_W. However, further increasing the iteration steps (e.g., $T = 4$ and $T = 5$) does not bring additional performance gains and even leads to slight metric degradation. This phenomenon may be attributed to potential over-smoothing or accumulated errors during the prolonged iterative process. Furthermore, the inference runtime increases linearly with T.

Table 3. Quantitative comparison of different T values.

Metrics	$T = 1$	$T = 2$	$T = 3$ (Ours)	$T = 4$	$T = 5$
Q_{MI}	0.3577	<u>0.3721</u>	**0.3732**	0.3717	0.3663
Q_{NCIE}	0.8053	<u>0.8055</u>	<u>0.8055</u>	0.8047	**0.8065**
$Q^{AB/F}$	0.2061	0.2184	**0.2199**	<u>0.2187</u>	0.2185
Q_P	0.0618	0.0660	**0.0755**	<u>0.0693</u>	0.0689
Q_{CB}	0.3216	0.3587	**0.3790**	<u>0.3768</u>	0.3673
Q_W	0.6192	<u>0.6221</u>	**0.6237**	0.6178	0.6133
Runtime (s)	**0.1425**	0.2201	0.3024	0.3768	0.4589

Therefore, taking both fusion quality and computational efficiency into consideration, we set $T = 3$ as the default configuration for our model.