RDF-MIG: A Robust Diffusion Framework for Masked Image Generation to Augment Semantic Segmentation and Change Detection

Supplementary Material

1. Supplementary experiments

1.1. Super-Resolution Comparison Experiments

In this subsection, we adopt the pretrained LDM model as the backbone of the diffusion-based generation module. We first train our RDF-MIG model and the ChangeDiff model on the Hi-CNA dataset, and then use each of them to generate synthetic training samples with a size twice that of the original Hi-CNA training set. Based on these generated datasets, we train the same downstream change detection models, SUN-Net and FC-Siam-conc, using different synthetic data configurations and finally compare their performance. The results are summarized in Fig. 2

Table 1. All methods are trained on the Hi-CNA dataset. In this experiment, our method trains the diffusion model using the proposed MCRD loss.

	SUN-Net			FC-Siam-conc		
	IOU	F1	Recall	IOU	F1	Recall
ChangeDiff	55.03	70.99	68.64	32.26	48.78	51.41
Ours	60.12	75.09	74.83	38.75	55.86	55.34
Ours+NIR	**61.81**	**76.40**	**75.55**	**46.29**	**63.30**	**60.27**

As shown in the table, the proposed RDF-MIG method leverages the FCF module to enable joint generation of feature images and masks in a three-channel setting, thereby achieving super-resolution data generation. In addition, RDF-MIG constructs a joint probability distribution over image and mask pixels during training, which effectively avoids the noise error accumulation problem commonly encountered in conditional control strategies (where an extra encoder–decoder network is required to model the relationship between every image–mask pixel pair). As a result, the generated image–mask pairs are better matched and more consistent. At a resolution of 512×512, the downstream change detection models trained on synthetic data generated by the proposed method significantly outperform those trained with the recently proposed ChangeDiff approach. In addition, the proposed method can effectively exploit the multispectral information in the dataset. When the downstream models are trained on multispectral synthetic data with an NIR band generated by our method, their performance is further improved. These experimental results demonstrate that the RDF-MIG framework is capable of generating high-quality super-resolution image–mask pairs and supports multispectral image generation.

1.2. Visualization of multispectral imagery

We present the multispectral images generated by our model and decompose them into RGB and NIR bands to evaluate its multispectral generation capability. The NIR band is visualized as a single-channel grayscale image after normalization, solely for intuitively illustrating its spatial distribution characteristics.

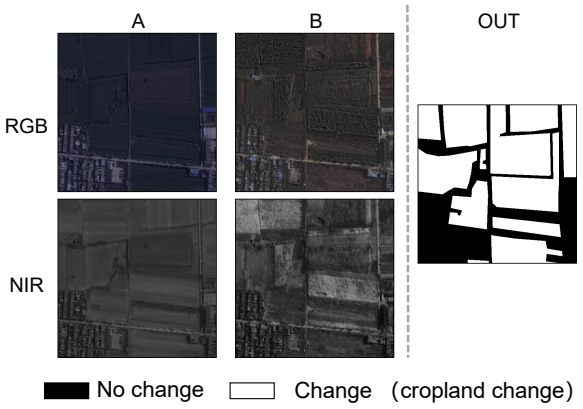

Figure 1. Visualization of synthesized multispectral imagery. The figure shows the pairs of multispectral images and corresponding change masks generated by the RDF-MIG framework trained on the Hi-CNA dataset.

From Fig. 1, we can see that the proposed RDF-MIG framework can effectively generate multispectral images including the NIR band, demonstrating its capability for multispectral image generation and overcoming the limitation of existing methods that can only generate RGB images.

1.3. Super-resolution visualization

As illustrated in Fig. 2, the proposed RDF-MIG framework exhibits strong super-resolution generation capability. On the Hi-CNA dataset, we instantiate RDF-MIG with DDPM and LDM backbones to synthesize image–mask pairs at resolutions of 128×128 and 512×512, respectively, and qualitatively visualize the generated results.

For fair visual comparison, images of different resolutions are uniformly resized to a common display size, enabling a more intuitive assessment of their effective resolution and detail sharpness without altering their underlying spatial structure.

From the Fig. 2, we observe that the 512×512 high-resolution images generated by the LDM-based RDF-MIG exhibit clearer textures and scene structures, showing a

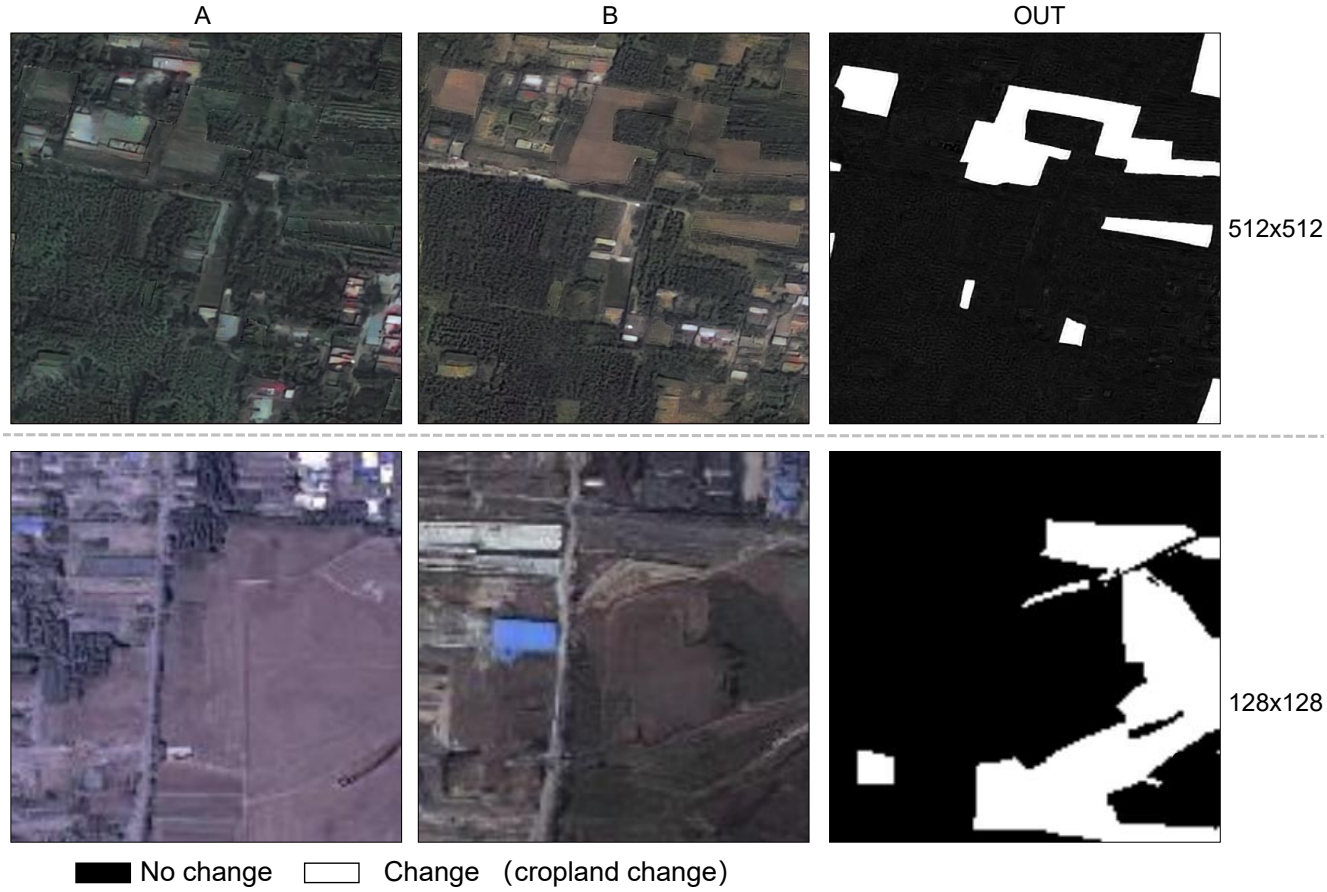

Figure 2. **Super-resolution generation results of RDF-MIG.** The RDF-MIG framework is instantiated with LDM and DDPM to generate image–mask pairs at resolutions of 512×512 and 128×128, respectively. For fair visual comparison, images of different resolutions are uniformly resized to a common display size. The high-resolution results (top) exhibit clearer textures, sharper structures, and smoother mask boundaries compared with the low-resolution results (bottom).

clear visual advantage over the 128×128 results produced by the DDPM-based model. Meanwhile, the masks generated by the high-resolution model have smoother and more regular boundaries, with noticeably finer edge details than those obtained from the low-resolution model.

The experimental results demonstrate that the proposed RDF-MIG method can be effectively combined with pre-trained diffusion models that support super-resolution, such as LDM, exhibiting strong generality and allowing flexible selection of diffusion backbones according to different requirements.

1.4. FCF Ablation Experiments

To further verify the necessity of the FCF as a core component of the RDF-MIG framework and its critical role in the joint generation of image–mask pairs, we conduct an ablation study on this module in this section.

As described in the main paper, the FCF module not only fuses images and masks, but also performs dimensionality reduction on image features. To highlight the effect of FCF, we design a control experiment without this module for the change detection task. Specifically, using the Hi-CNA dataset as an example, the control group does not apply FCF; instead, we directly concatenate each paired bi-temporal remote sensing image $\boldsymbol{x}_{A,i} \in \boldsymbol{R}^{H\times W\times 3}$ and $\boldsymbol{x}_{B,i} \in \boldsymbol{R}^{H\times W\times 3}$ with the corresponding change mask $\boldsymbol{m}_{cd,i} \in \boldsymbol{R}^{H\times W\times 1}$ along the channel dimension to construct a fused feature map $\boldsymbol{z}_i \in \boldsymbol{R}^{H\times W\times 7}$ with 7 channels, where channels 1–3 correspond to the RGB image at time A, channels 4–6 to the RGB image at time B, and channel 7 to the change mask. Based on this, we then perform generative modeling using the same DDPM architecture and training pipeline as in RDF-MIG. During the generation phase, the newly generated fused feature maps $\tilde{\boldsymbol{z}}_i \in \boldsymbol{R}^{H\times W\times 7}$

are split, according to the above channel definition, into two RGB images at different times and the corresponding change mask, forming the control synthetic dataset $\boldsymbol{D}^*$.

It is worth noting that, apart from removing the FCF module, the control group shares exactly the same training procedure and hyperparameter settings as the standard group. Since the increased number of channels incurs additional memory cost, we adopt gradient accumulation to ensure that the effective batch size is identical in both experiments. Finally, we use the standard-group synthetic dataset $\boldsymbol{D}$ and the control-group synthetic dataset $\boldsymbol{D}^*$ to train downstream change detection models with the same architecture, and compare their performance to indirectly assess the quality difference between the two types of generated data. The experimental results are shown in Tab. 2.

Table 2. **Ablation study on the FCF module**. All methods are evaluated on the Hi-CNA dataset. In this experiment, apart from whether the FCF processing pipeline is applied, the remaining training procedure (including the loss functions and all hyperparameter settings) is kept exactly the same. All methods operate on images with a resolution of 128×128 pixels, and the diffusion model is implemented using DDPM, with the specific settings identical to those in the experiments described in Section 5.2 of the main paper.

	SUN-Net			STANet		
	IOU	F1	Recall	IOU	F1	Recall
RDF-MIG(without FCF)	20.11	33.48	35.03	17.84	30.28	38.91
RDF-MIG	**50.00**	**66.67**	**69.21**	**45.72**	**62.75**	**67.31**

As shown in the table, removing the FCF module and directly concatenating all inputs leads to an excessive number of channels, making it difficult for the diffusion model to effectively learn the joint distribution between the bi-temporal images and the change mask. As a result, the quality of the generated data degrades and it can hardly provide useful training samples for downstream models. In contrast, introducing the FCF module not only significantly reduces the channel dimensionality, but also preserves the correspondence between images and masks and constructs a joint image–mask distribution that is more suitable for the diffusion model to learn. Therefore, the FCF module is crucial for the joint generation of image–mask pairs.

To further demonstrate the reliability of the experimental results, we present a visual comparison of the data generated by the RDF-MIG framework with and without the FCF module. The results are shown in Fig. 3.

From the results in Fig. 3, we can observe that when the FCF module is removed, the generated images are heavily blurred, making it almost impossible to visually identify the land-cover categories and their spatial structures. In contrast, the data generated by the standard group with the FCF module exhibit visual quality that is nearly indistinguishable from the real dataset.

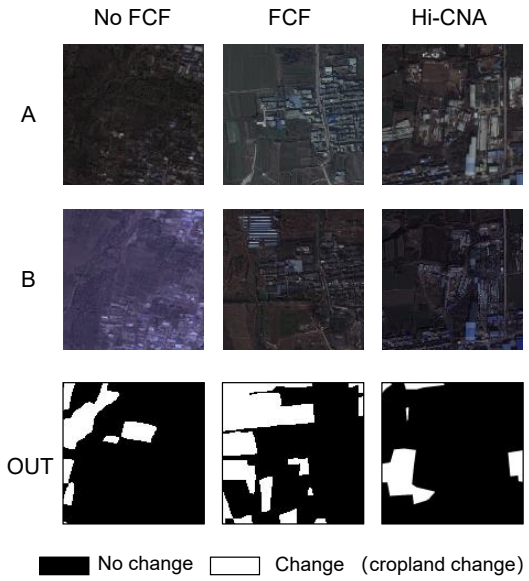

Figure 3. **Visual comparison of change detection results on the Hi-CNA dataset**. Since adding the NIR band without the FCF module would result in an excessive number of channels and computational requirements beyond our available resources, this experiment is conducted using only RGB images. The figure shows training samples from the Hi-CNA dataset, along with the images and corresponding change masks generated by the control group without the FCF module and the standard group with the FCF module.

Finally, it should be noted that within the proposed RDF-MIG framework, the FCF module may not represent the most complex or cutting-edge design in terms of feature reduction and related operations. However, its significance lies in providing a scheme that simultaneously achieves effective dimensionality reduction and relationship modeling, enabling the construction of a joint image–mask probability distribution that is well suited for diffusion model training, and naturally extending to multi-spectral image generation. The experimental results in this section not only intuitively demonstrate the practical role of the FCF module, but also indirectly explain why existing methods struggle to support the joint generation of multi-spectral image–mask pairs, namely that an excessive number of channels makes it difficult for diffusion models to learn effective representations. Therefore, the contribution of the proposed FCF module does not lie in introducing a complex new network architecture, but in providing a novel and effective design paradigm for the joint generation of multi-spectral image–mask pairs. In future work, we will further explore more efficient and accurate FCF–decoder combinations.

2. Theoretical Analysis of MCRD Losses

2.1. Derivation of Diffusion Models

Before delving into MCRD in depth, it is necessary to first provide a more detailed explanation of the fundamental principles of diffusion models.

Forward process Given any clear image $\boldsymbol{x}_0$ and gradually added Gaussian noise to it. After a sufficient number of steps, $\boldsymbol{x}_0$ will become an image resembling pure Gaussian noise. This step-by-step noise addition process is referred to as the forward process and can be represented as follows:

Assuming we start from $\boldsymbol{x}_0$ and add Gaussian noise in a total of T steps, the image $\boldsymbol{x}_t$ at any given time t can be represented as:

$$\boldsymbol{x}_t = \sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \sqrt{1-\alpha_t}\varepsilon_t \tag{1}$$

where $t \in \{0, 1, 2, ..., T\}$ represents the time step of the diffusion process, $\boldsymbol{x}_t$ is the image after t steps, $\alpha_t = (1-\beta_t) \in (0,1)$ is a scaling factor, $\varepsilon_t \sim N_\theta(0, \boldsymbol{I})$ is the noise image randomly sampled from a standard Gaussian distribution at step t, and $N_\theta\left(\mu, \sigma^2\boldsymbol{I}\right)$ denotes a multivariate normal distribution where each element is normally distributed with mean μ and variance σ^2.

Based on Equation (1), we can further derive a formula for directly calculating the image $\boldsymbol{x}_t$ at any time step t from the initial image $\boldsymbol{x}_0$:

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon \tag{2}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, $\varepsilon \sim N_\theta(0, \boldsymbol{I})$ is noise randomly sampled from a standard Gaussian distribution, and it is related to ε_t. In summary, we can understand it as follows: under the given condition $\boldsymbol{x}_0$, the outcome $\boldsymbol{x}_t$ at the end of the forward process is uniquely determined by the fixed noise ε.

Reverse Process Before further introducing the MCC loss function, it is necessary to provide the following assumption:

Assumption 1: In the forward propagation process, we assume that t is sufficiently large, so that $\boldsymbol{x}_t$ can be approximated as standard Gaussian noise.

Based on ***Assumption 1*** and the derivation of the forward process, we can consider $\hat{\boldsymbol{x}}_t$ as Gaussian noise sampled from a standard Gaussian distribution. The key to reverse denoising in diffusion models is learning how to denoise the noisy image $\hat{\boldsymbol{x}}_t$ to generate $\hat{\boldsymbol{x}}_{t-1}$ in the reverse direction. Furthermore, the reverse process of the diffusion model can be understood as starting from a noisy image $\hat{\boldsymbol{x}}_t$ sampled from a standard Gaussian distribution, progressively denoising in the reverse direction, and eventually producing new data $\hat{\boldsymbol{x}}_0$ with a distribution similar to the training data $\boldsymbol{x}_0$. This can be represented by the following equation:

$$\min \sum_{t=1}^{T} D_{KL}\left[p\left(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0\right)||p\left(\hat{\boldsymbol{x}}_{t-1}|\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{x}}_0\right)\right] \tag{3}$$

where $D_{KL}\left[p_1||p_2\right]$ represents the KL divergence between probability distributions p_1 and p_1, and $p\left(.\right)$ denotes a probability distribution. Therefore, it is necessary to train a model that can accurately predict the true probability distribution $p\left(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0\right)$.

Since the forward diffusion process is a Gaussian process, the conditional distribution of the reverse process will also be Gaussian. Therefore, using Bayes' theorem, we can derive:

$$p\left(\hat{\boldsymbol{x}}_{t-1}|\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{x}}_0\right) = N_\theta\left(\mu_t\left(\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{x}}_0\right), \sigma^2\boldsymbol{I}\right) \tag{4}$$

where

$$\sigma^2 = \frac{\beta_t\left(1-\bar{\alpha}_{t-1}\right)}{\alpha_t\left(1-\bar{\alpha}_{t-1}\right)+\beta_t} \tag{5}$$

and

$$\mu_t\left(\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{x}}_0\right) = \frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t}\hat{\boldsymbol{x}}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\hat{\boldsymbol{x}}_0 \tag{6}$$

Since the ultimate goal is for the predicted probability distribution $p\left(\hat{\boldsymbol{x}}_{t-1}|\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{x}}_0\right)$ to match the true distribution $p\left(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0\right)$, this can be equivalently achieved by making the means of the probability distributions equal $\hat{\mu}_t\left(\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{x}}_0\right) = \mu_t\left(\boldsymbol{x}_t, \boldsymbol{x}_0\right)$. However, in the reverse process, $\hat{\boldsymbol{x}}_0$ is unknown. Therefore, the following calculations are also required:

Using Equation (2) and Equation (6) derived from the forward process, we can get

$$\begin{cases} \mu_t\left(\boldsymbol{x}_t, \boldsymbol{x}_0\right) = \dfrac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t}\boldsymbol{x}_t + \dfrac{\beta_t\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_t}\boldsymbol{x}_0 \\ \boldsymbol{x}_0 = \dfrac{1}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_{t-1}}\right)\varepsilon \end{cases} \tag{7}$$

Next, combining Equation (7), can obtain:

$$\begin{cases} \mu_t\left(\boldsymbol{x}_t, \boldsymbol{x}_0\right) = A_{\bar{\alpha}_t}\boldsymbol{x}_t + B_{\bar{\alpha}_t, \beta_t}\varepsilon \\ \hat{\mu}_t\left(\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{x}}_0\right) = A_{\bar{\alpha}_t}\hat{\boldsymbol{x}}_t + B_{\bar{\alpha}_t, \beta_t}\hat{\varepsilon} \end{cases} \tag{8}$$

where $A_{\bar{\alpha}_t}$ is a constant related to $\bar{\alpha}_t$, and $B_{\bar{\alpha}_t, \beta_t}$ is a constant related to both $\bar{\alpha}_t$ and β_t. From Equation (8), we can conclude that it is sufficient to train a model that, given the input $\hat{\boldsymbol{x}}_t$, can accurately predict the corresponding noise $\hat{\varepsilon}$. In this way, the goal of Equation (3) can be indirectly achieved. Through the forward process, we obtained a large number of sample pairs $(\boldsymbol{x}_t, \varepsilon)$. Therefore, traditional diffusion models are trained directly by simply minimizing the mean squared error between the predicted and actual noise:

$$\min E_{\mathrm{t}, \boldsymbol{x}_0, \varepsilon}\left[\left\|f_\theta\left(\boldsymbol{x}_t, t\right) - \varepsilon\right\|^2\right] \tag{9}$$

The above is the design principle of the DDPM model.

2.2. Limitations of Existing Diffusion Losses

Building on the theoretical derivation of diffusion models in Section 2.1, we now proceed to point out the limitations of existing diffusion model losses.

The ultimate goal of training a diffusion model is to find the true probability distribution $\sum_{t=1}^{T} p\left(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0\right)$ corresponding to all data in the dataset. This objective can ultimately be expressed as minimizing the MSE loss function:

$$\min E_{\mathrm{t},\mathrm{x}_0,\varepsilon}\left[\left\|f_\theta\left(\mathrm{x}_t, t\right)-\varepsilon\right\|^2\right] \tag{10}$$

By taking the derivative of the MSE loss function in Equation (10), we can obtain the gradient expression $\boldsymbol{g}_{MSE,\theta}$ of the algorithm as follows:

$$\begin{aligned}\boldsymbol{g}_{MSE,\theta} &= \partial E_{\mathrm{t},\boldsymbol{x}_0,\varepsilon}\left[\left\|f_\theta\left(\boldsymbol{x}_t, t\right)-\varepsilon\right\|^2\right]/\partial\theta \\ &= 2E_{\mathrm{t},\boldsymbol{x}_0,\varepsilon}\left[e\frac{\partial e}{\partial\theta}\right]\end{aligned} \tag{11}$$

where, $e = f_\theta\left(\boldsymbol{x}_t, t\right)-\varepsilon$ represents the error between the predicted noise and the true noise, and θ denotes the weight parameters of the model that need to be updated.

However, this training approach is overly idealized. In practice, datasets inevitably contain noisy data, such as labeling errors and inconsistent image styles. For example, the labeling accuracy of the HRSCID dataset is only between 80% and 85%. When noisy data exists in the dataset, training a diffusion model based on the MSE cost function will shift the intended training objective. This is due to the fact that the target noise ε for model training no longer originates from the ideal probability distribution $\sum_{t=1}^{T} p\left(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0\right)$, but is instead sampled from an erroneous mixed Gaussian distribution $\sum_{t=1}^{T} p^*\left(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0\right)$. The definition of $\sum_{t=1}^{T} p^*\left(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0\right)$ is as follows:

$$\begin{aligned}&\sum_{t=1}^{T} p^*\left(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0\right) \\ &= \lambda\sum_{t=1}^{T} p\left(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0\right) + (1-\lambda)\sum_{t=1}^{T} p^{**}\left(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0\right)\end{aligned} \tag{12}$$

where $\sum_{t=1}^{T} p\left(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0\right)$ is the probability distribution corresponding to real data, $p^{**}\left(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0\right)$ is the probability distribution corresponding to noisy data, and $0 < \lambda < 1$ is a weight parameter determined by the proportion of noisy data and real data in the dataset.

The reason for this outcome is that the training objective of the MSE loss function is to push the model to minimize the error across all data points as much as possible, thereby further minimizing the overall expected error:

$$\min E_{\mathrm{t},\boldsymbol{x}_0,\varepsilon}\left[\left\|f_\theta\left(\boldsymbol{x}_t, t\right)-\varepsilon\right\|^2\right] \tag{13}$$

Due to the presence of misdata ε^*, when the algorithm encounters misdata during training, it forces the model to update its parameters, causing the predicted output $f_\theta\left(\boldsymbol{x}_t, t\right)$ to gradually approach the ε^*. Observing Equation (11), we can see that the model is originally supposed to converge toward the true probability distribution. Ideally, both the loss function and the gradient should gradually decrease. However, due to the presence of anomalous data, the prediction error e unexpectedly increases, which in turn causes the gradient $\boldsymbol{g}_{MSE,\theta}$ to increase as well, ultimately leading the model parameters to update in the wrong direction. Furthermore, since the ultimate goal of the MSE loss function is to minimize the expected error between the model's predicted noise and the noise corresponding to all data, it eventually forces the model to train toward a mixed distribution that lies between the true distribution and the erroneous distribution, of a diffusion model designed with the MSE loss function will inevitably deteriorate. This ultimately results in the model generating low-quality or erroneous data when producing outputs. as shown in Equation (12). Therefore, as long as noisy data is present in the dataset, the performance will inevitably deteriorate.

Compared with the MSE loss, the recently proposed L1 loss for improving the robustness of diffusion models modify the gradient in the large-error regime, changing it from the form in Eq. (11) to:

$$\begin{aligned}\mathrm{g}_{L1,\theta} &= \partial E_{t,\mathrm{x}_0,\varepsilon}\left[\left|f_\theta\left(x_t, t\right)-\varepsilon\right|\right]/\partial\theta \\ &= E_{t,\mathrm{x}_0,\varepsilon}\left[sign\left(e\right)\frac{\partial e}{\partial\theta}\right]\end{aligned} \tag{14}$$

In addition, recent studies extend the L1 loss by introducing the Huber loss, which simultaneously preserves the fine-grained optimization capability of the MSE loss for small errors and the robustness of the L1 loss for large errors. Specifically, when the magnitude of the error is below a predefined threshold δ, the Huber loss reduces to the quadratic form of the MSE loss; when the error exceeds δ, it switches to the linear form of the L1 loss.

However, as shown in Eq. (12) and Eq. (14), when the model encounters noisy samples during training, the target distribution it attempts to fit has already shifted. Therefore, even if the loss is switched to L1, its effect is limited to reducing the magnitude of parameter updates induced by noisy samples (i.e., yielding smaller gradients than MSE in the large-error regime), but it still cannot prevent the model parameters from being updated in an incorrect direction under the influence of these samples.

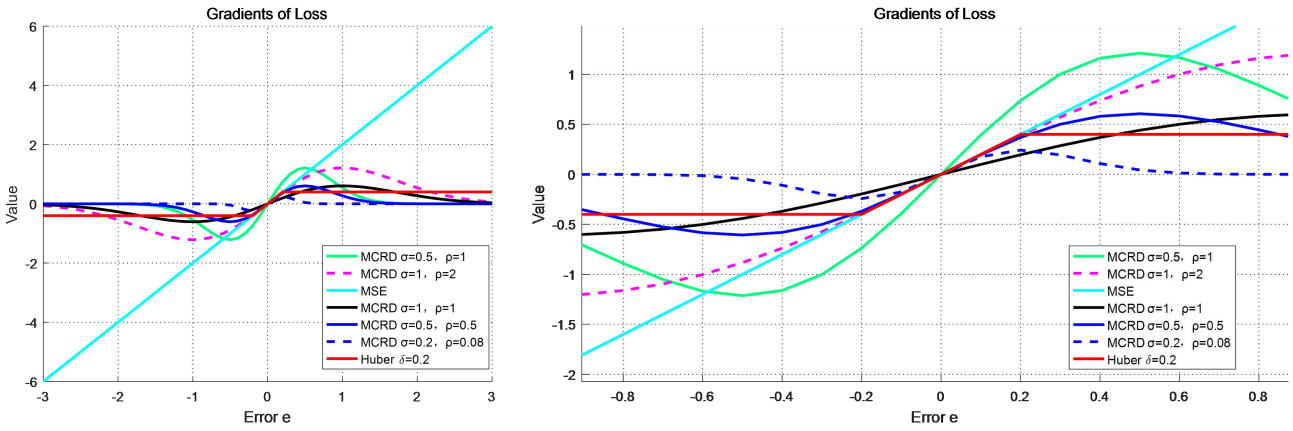

Figure 4. **Gradient curves for different losses.** We plot the gradient curves of different loss functions, with a particular focus on comparing the gradients of the MCRD loss under different kernel widths σ and scaling parameters ρ. The left panel shows the global gradient curves, while the right panel presents a zoomed-in local view, making it easier to clearly observe how different parameter settings affect the shape and behavior of the gradients.

2.3. MCRD Loss

The previous subsection has explained the limitations of existing loss functions used for diffusion model training. In what follows, we provide a detailed theoretical analysis of the advantages of the proposed MCRD loss, as well as the influence of its hyperparameters and the corresponding selection rules.

Based on Eqs. (5)–(7) in the main paper, we derive the gradient of the MCRD loss as:

$$\boldsymbol{g}_{\text{mcrd},\theta} = \boldsymbol{E}_{t,\boldsymbol{x}_0,\varepsilon}\left[\rho(1/\sigma^2)\exp\left(-\frac{e^2}{2\sigma^2}\right)e\frac{\partial e}{\partial\theta}\right] \tag{15}$$

In addition, we propose an MSE-consistency calibration (i.e., setting $\rho = 2\sigma^2$) to endow the MCRD loss with robustness while ensuring that it maintains training accuracy comparable to that of the MSE loss. Based on Eq. (15), and under the MSE-consistency calibration, we derive in the main paper the basic selection rule for the kernel width σ as follows:

$$\frac{3{e_{\max}}^2}{2\sigma^2} \le \beta \tag{16}$$

where $e_{\max}$ denotes the error range within which we expect the MCRD gradient to closely match that of the MSE loss, i.e., for $e \in (-e_{\max}, e_{\max})$.

Next, we provide a detailed derivation of how the kernel-width parameter σ affects the gradient of the MCRD loss. Under the constraint $\rho = 2\sigma^2$, Eq. (15) can be rewritten as:

$$\boldsymbol{g}_{\text{mcrd},\theta} = \boldsymbol{E}_{t,\boldsymbol{x}_0,\varepsilon}\left[2\exp\left(-\frac{e^2}{2\sigma^2}\right)e\frac{\partial e}{\partial\theta}\right] \tag{17}$$

To simplify the derivation, we ignore the common factors that are independent of θ and rewrite Eq. (15) as a function that depends only on weight θ:

$$h\left(e\right) = 2e\exp\left(-\frac{e^2}{2\sigma^2}\right) \tag{18}$$

Further, taking the derivative of Eq. (18) can obtain:

$$h^*\left(e\right) = 2\exp\left(-\frac{e^2}{2\sigma^2}\right)\left(1-\frac{e^2}{\sigma^2}\right) \tag{19}$$

From Eq. (19), we observe that when $|e| < \sigma$, we have $h^*(e) > 0$, and the gradient increases as $|e|$ grows. In contrast, when $|e| > \sigma$, we have $h^*(e) < 0$, and the gradient starts to decrease as the error continues to increase. Furthermore, by taking the second derivative of $h(e)$, we obtain:

$$h^{**}\left(e\right) = -2\exp\left(-\frac{e^2}{2\sigma^2}\right)\frac{e^2}{\sigma^2}\left(3-\frac{e^2}{\sigma^2}\right) \tag{20}$$

Based on Eqs. (19) and (20), we observe that when $|e| = \sigma$, the gradient update magnitude reaches an extremum, marking the turning point at which the MCRD gradient transitions from increasing with $|e|$ to decreasing with $|e|$. In the interval $\sigma < |e| < \sqrt{3}\,\sigma$, the first derivative is negative and the second derivative is also negative, so the gradient not only decreases monotonically but does so with an increasingly faster decay rate. When $|e| > \sqrt{3}\,\sigma$, the gradient continues to decay, but the decay rate slows down and the gradient value is already close to zero, thereby effectively suppressing the interference of heavy-tailed noise on the update of model parameters. Fig. 4 compares the gradient curves of different loss functions, with a particular emphasis on illustrating how the MCRD gradient varies under different kernel widths σ and scaling parameters ρ. From the figure, we can clearly observe that the behavior of the curves is consistent with our theoretical derivations, which

validates the correctness of our analysis.

In summary, the kernel width σ determines the boundary at which the MCRD loss transitions from the MSE-like fine-fitting region to the robustness region that suppresses noise. In practical applications, an appropriate value of σ can be chosen by jointly considering the desired fine-update error range $e_{\max}$, the allowable deviation rate β, and the error scale at which robustness is expected to take effect. This allows balancing training accuracy and robustness against noise.

3. Image–Mask Generation Feasibility

This section supplements the main text with a theoretical feasibility analysis of the joint generation of image–mask pairs.

Taking the change detection data generation task as an example, two multispectral images and the change mask in dataset $\boldsymbol{D}$ are passed through the FCF module for dimensionality reduction and fusion, resulting in the latent feature image $\boldsymbol{z}_{i,0} \in \boldsymbol{R}^{H\times W\times 3}$. Among them, the three channels of the feature map $\boldsymbol{z}_{i,0}$ correspond to the dimension-reduced image at time A, the dimension-reduced image at time B, and the change mask, respectively.

Next, progressively add Gaussian noise to $\boldsymbol{z}_{i,0}$. Similarly, after a sufficient number of steps, $\boldsymbol{z}_{i,0}$ will become a three-channel image that approximates Gaussian noise $\boldsymbol{z}_{i,t}$. The forward noise addition process can be represented as:

$$\begin{cases} \boldsymbol{z}_{i,t} = \sqrt{\alpha_t}\boldsymbol{z}_{i,t-1} + \sqrt{1-\alpha_t}\varepsilon_{z,t} \\ \boldsymbol{z}_{i,t} = \sqrt{\bar{\alpha}_t}\boldsymbol{z}_{i,0} + \sqrt{1-\bar{\alpha}_t}\varepsilon_z \end{cases} \tag{21}$$

where $(\varepsilon_z, \varepsilon_{z,t}) \sim N_\theta\left(0, \boldsymbol{I}_z\right)$ and $\boldsymbol{I}_z \in \boldsymbol{R}^{H\times W\times 3}$.

In an ideal scenario, all images in dataset $\boldsymbol{D}$ are high-quality and perfectly aligned with their corresponding masks. Under these conditions, processing each image–mask pair with the FCF module yields a new dataset $\boldsymbol{M}$:

$$\boldsymbol{M} = \left\{\boldsymbol{z}_{i,0} | \boldsymbol{z}_{i,0} \in \boldsymbol{R}^{H\times W\times 3}\right\}, i = 1, 2, ..., N \tag{22}$$

In addition, all data in dataset $\boldsymbol{M}$ can be regarded as generated by starting from the same standard Gaussian distribution and undergoing t steps of reverse denoising to obtain. Therefore, the training objective can be expressed as Equation (23):

$$\min \sum_{t=1}^{T} D_{KL}\left[p\left(\boldsymbol{z}_{i,t-1} | \boldsymbol{z}_{i,t}, \boldsymbol{z}_{i,0}\right) || p\left(\hat{\boldsymbol{z}}_{t-1} | \hat{\boldsymbol{z}}_t, \hat{\boldsymbol{z}}_0\right)\right] \tag{23}$$

where $\hat{\boldsymbol{z}}_0$ is the concatenated image generated by the model, and $\hat{\boldsymbol{z}}_t$ is a noise image randomly sampled from a standard Gaussian distribution. Since we assume that the data in $\boldsymbol{M}$ is ideal, we have:

$$p\left(\boldsymbol{z}_{i,t-1} | z_{i,t}, \boldsymbol{z}_{i,0}\right) = p\left(\boldsymbol{z}_{j,t-1} | \boldsymbol{z}_{j,t}, \boldsymbol{z}_{j,0}\right), (i \neq j) \in N \tag{24}$$

Therefore, when we have all the ideal probability distributions $\sum_{t=1}^{T} p\left(\boldsymbol{z}_{i,t-1} | \boldsymbol{z}_{i,t}, \boldsymbol{z}_{i,0}\right)$, performing reverse denoising from this distribution will generate new images $\hat{\boldsymbol{z}}_0$ that satisfy:

$$\hat{\boldsymbol{z}}_0 = \hat{\boldsymbol{x}}_{A,0} \oplus \hat{\boldsymbol{x}}_{B,0} \oplus \hat{\boldsymbol{m}}_0 \tag{25}$$

where $\oplus$ is the concatenation operation.

The above process theoretically proves that the method proposed in this paper can generate corresponding mask labels while generating new images.