# Learning to Describe Implicit Changes: Noise-robust Pre-training for Image Difference Captioning

**Zixin Guo**[1], **Jiayang Sun**[2], **Tzu-Jui Julius Wang**[3],
**Abduljalil Radman**[1], **Selen Pehlivan**[4], **Min Cao**[2*], **Jorma Laaksonen**[1]
[1] Aalto University, [2] Soochow University, [3] Zenseact AB
[4] VTT Technical Research Centre of Finland
zixin.guo@aalto.fi, mcao@suda.edu.cn

## Abstract

Image Difference Captioning (IDC) methods have advanced in highlighting subtle differences between similar images, but their performance is often constrained by limited training data. Using Large Multimodal Models (LMMs) to describe changes in image pairs mitigates data limits but adds noise. These change descriptions are often coarse summaries, obscuring fine details and hindering noise detection. In this work, we improve IDC with a noise-robust approach at both data and model levels. We use LMMs with structured prompts to generate fine-grained change descriptions during data curation. We propose a Noise-Aware Modeling and Captioning (NAMC) model with three modules: Noise Identification and Masking (NIM) to reduce noisy correspondences, Masked Image Reconstruction (MIR) to correct over-masking errors, and Fine-grained Description Generation (FDG) to produce coherent change descriptions. Experiments on four IDC benchmarks show that NAMC, pre-trained on our large-scale data, outperforms streamlined architectures and achieves competitive performance with LLM-finetuned methods, offering better inference efficiency.

## 1 Introduction

Image Difference Captioning (IDC) involves discerning fine-grained changes between two similar images and generating *change descriptions* that summarize these changes (Jhamtani and Berg-Kirkpatrick, 2018). Unlike general image captioning task (Vinyals et al., 2015), which describes the content of a single image, IDC requires fine-grained semantic understanding to accurately detect and articulate subtle changes between two visually and semantically similar images. The ability of this task to model these nuanced changes proves invaluable in various applications, like medical case comparison (Liu et al., 2021; Beddiar et al., 2023), remote sensing monitoring (Hoxha et al., 2022; Chang and Ghamisi, 2023), and security monitoring (Jhamtani and Berg-Kirkpatrick, 2018).

Recently, IDC approaches have made long-term progress (Tu et al., 2021a; Yao et al., 2022). However, their performance is often constrained by limited training data. This limitation arises because acquiring IDC training data—comprising similar image pairs and their corresponding change descriptions—is particularly challenging. The process heavily relies on manual annotations, making data curation both labor-intensive and time-consuming. Recent efforts (Jiao et al., 2024; Hu et al., 2024) have attempted to address this challenge by leveraging Large Multi-Modal models (LMMs) to generate change descriptions for pre-collected image pairs (Brooks et al., 2023). However, these generated descriptions often introduce noise in the form of obvious, irrelevant, or redundant details, resulting in weak annotations compared to those created by human annotators, undermining model performance. In particular, such descriptions tend to summarize changes in a coarse manner, failing to capture fine-grained details from both images (Hu et al., 2024). This makes it challenging to identify and handle noisy correspondences between images and descriptions during subsequent model training, especially when considering noise-aware learning.

To this end, this work focuses on noise-robust image difference captioning at both the data and model levels, as depicted in Figure 1. At the data level, we aim to leverage LMMs to generate *fine-grained implicit-change descriptions* for similar image pairs, which provides the critical data foundation required for enabling model-level noise identification mechanisms. At the model level, we propose NAMC: a *Noise-Aware Modeling and Captioning* model designed to mitigate cross-modal noisy correspondences introduced by the generated data. NAMC incorporates three trainable modules: a Noise Identification and Masking (NIM) module

* Corresponding author



*Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10125–10145
November 4-9, 2025 

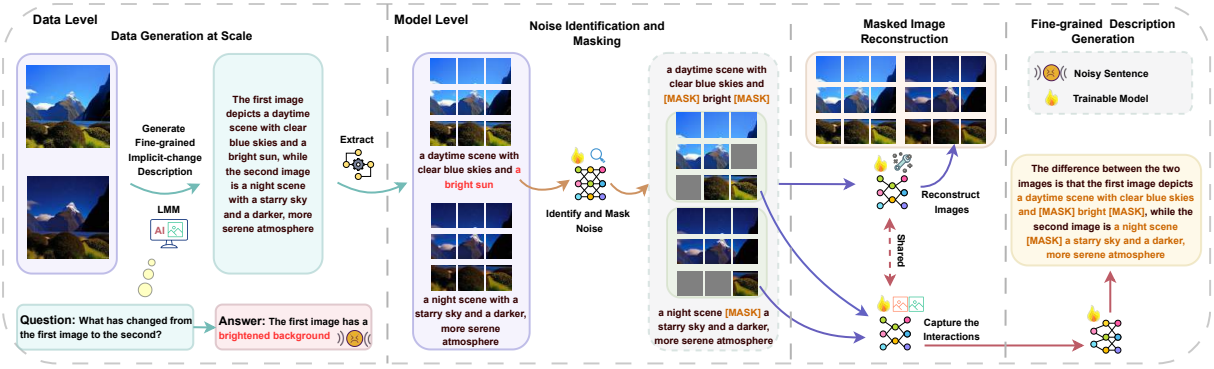



Figure 1: The skeleton of our proposed noise-robust image difference captioning, which includes fine-grained implicit-change descriptions at the data level and a noise-aware model at the model level.

to mitigate cross-modal noisy correspondences, a Masked Image Reconstruction (MIR) module to mitigate over-masking errors on images and enhance image comprehension, and a Fine-grained Description Generation (FDG) module to enable model to generate accurate change descriptions.

***Data Level.*** We curate the pre-training data generation process by using structured prompts to guide LMMs in generating descriptions that capture fine-grained differences between two images. Specifically, we aim to generate detailed descriptions for each image at a granular level, such as "The first image depicts a daytime scene with ..., while the second image is a night scene with ...". These descriptions implicitly capture changes between the images—for instance, the phrase "change it to night" can be inferred through the contrast between the two descriptions. Moreover, these descriptions can be explicitly aligned with corresponding image regions, facilitating noise-robust learning in subsequent model training.

***Model level.*** We develop three modules in the proposed NAMC. NIM module uses dual encoders—with contrastive loss—to project images and their corresponding descriptions into a shared space. Within this space, cross-modal token relevance computation guides a Noise Masking (NM) strategy to generate masks for both modalities. These masks filters noise from both images and descriptions, producing denoised images and descriptions. MIR module mitigates potential over-masking—relevant information is mistakenly masked—by reconstructing missing image content through a masked image encoder, while simultaneously modeling cross-image interactions that ignore noisy regions. This not only compensates for masking errors but also captures cross-image relationship modeling. FDG module leverages the cross-image relationships learned by MIR and employs a multi-modal decoder for the generation of change descriptions. It is pre-trained using the combined denoised description as supervision, enabling description of subtle inter-image distinctions through joint visual-textual decoding.

The main contributions of this work are summarized as: (1) we explore a noise-robust image difference captioning, using LMMs to generate large-scale change descriptions while mitigating the impact of noise from these generated descriptions during pre-training. (2) We introduce a pre-training dataset generation method that automatically produces fine-grained implicit-change descriptions, capturing subtle and implicit differences between image pairs. (3) We propose a noise-aware modeling and captioning model to address cross-modal noisy correspondences and enhance the comprehension of images and descriptions during pre-training. (4) Experiments on four IDC datasets show that our proposed NAMC achieves competitive performance with the LLM-finetuned state-of-the-arts, while maintaining superior inference efficiency.

## 2 Related Work

Within the domain of multi-modality learning (Guo et al., 2023; Kainulainen et al., 2024; Radman and Laaksonen, 2025), Image Difference Captioning (IDC) was first introduced by Jhamtani and Berg-Kirkpatrick (2018) and refined by Park et al. (2019) with distractors like viewpoint or brightness changes. Both followed a single-step training from scratch paradigm including an image encoder and a text decoder, typically with fewer parameters. Leveraging this paradigm, subsequent works mitigated the impact of distractors (Kim et al., 2021; Tu et al., 2024a) and concentrated on fine-grained difference recognition (Tu et al., 2023c, 2024b), providing a thorough exploration of the IDC task.

On the other hand, recent approaches have

adopted a two-step training that either (1) integrates auxiliary tasks alongside the main IDC objective, or (2) implements a pre-alignment stage prior to caption generation, as opposed to employing end-to-end training from scratch. Hosseinzadeh and Wang (2021) introduced an additional retrieval task as an auxiliary objective to enhance the performance of image difference captioning. Both Yao et al. (2022) and Guo et al. (2022) proposed a pretraining phase to align the image pairs with change descriptions before performing difference captioning. Particularly, Yao et al. (2022) used a cross-modal Transformer architecture to jointly process the images and captions, leveraging multiple contrastive learning tasks. Guo et al. (2022) utilized a CLIP-like architecture, aligning the [CLS] token from the encoder output with image-text retrieval objectives.

With the advent of LMMs (Peng et al., 2025; Jiang et al., 2025), research focus has shifted toward applying these models to IDC and fully leveraging their rich prior knowledge. Lu et al. (2023) and Zhang et al. (2024) partially fine-tuned BLIP2 (Li et al., 2023) model for IDC task, with Zhang et al. (2024) specifically introducing retrieval-augmented generation to improve captioning. Other methods (Black et al., 2024; Hu et al., 2024; Jiao et al., 2024) utilized knowledge from LMMs to generate additional training data for fine-tuning. Despite promising results, handling noise interference is still a challenge for models using LMMs to generate additional data. Inspired by the success of noise-robust techniques across various domains (Kang et al., 2023; Fu et al., 2024; Tan et al., 2024; Huang et al., 2024), we propose a noise-aware IDC model to address cross-modal noisy correspondences in LMM-generated data.

## 3 Method

Figure 2 shows our proposed noise-robust image difference captioning framework, which begins with generating fine-grained implicit-change descriptions using LMMs on curated image pairs. We then introduce the NAMC model, composed of three modules, NIM for noise mitigation, MIR for reconstruction, and FDG for change description generation, followed by the pre-training strategies.

### 3.1 Data Generation at Scale

Manual obtaining of image pairs and annotation of their change descriptions is time- and labor-intensive; therefore, we rely on automatic data generation in our study. This process involves (1) collecting similar image pairs and (2) generating change descriptions for them.

#### 3.1.1 Similar Image Pairs

We collect an image dataset comprising approximately 770K synthetic and real image pairs. Of these, 313K synthetic pairs come from an existing dataset ("clip-filtered-dataset" version) provided by InstructPix2Pix (Brooks et al., 2023). For real pairs, we leverage two sources: 8K pairs from the existing Spot-the-Diff dataset (Jhamtani and Berg-Kirkpatrick, 2018), and 450K pairs from the CC3M dataset (Sharma et al., 2018). Further details on the image collection and our similar image pair construction pipeline are provided in Appendix D.

#### 3.1.2 Implicit-change Descriptions

Recent works prompt LMMs to generate change descriptions for a collection of image pairs (Brooks et al., 2023), often resulting in summaries of changes. However, these descriptions, generated via typical prompts, often introduce noise by including extraneous details or by obscuring key differences between the images. More importantly, this noise propagates to the model level, making it harder to identify noisy correspondences between the image pairs and their corresponding change descriptions. To address these challenges, we introduce the fine-grained implicit-change description—a format designed to capture key differences between two images, which is essential for the IDC task. The fine-grained implicit-change description follows the structured template:

> *The difference between the two images is that the first image {descriptive verb} [object description], while the second image {descriptive verb} [object description].*

In the template, "*{descriptive verb}*" is a placeholder for a replaceable verb, e.g. describes, chosen from a predefined list provided in Appendix D. Similarly, "*[object description]*" is a placeholder for LMMs to fill in specific details of objects undergoing change, thereby focusing on the differences between the two images.

Unlike directly summarized change descriptions that explicitly describe the change between two images from Hu et al. (2024), our format preserves fine-grained details from both images, allowing the change to be inferred. Furthermore, this descriptive format enables the extraction of "*[object*

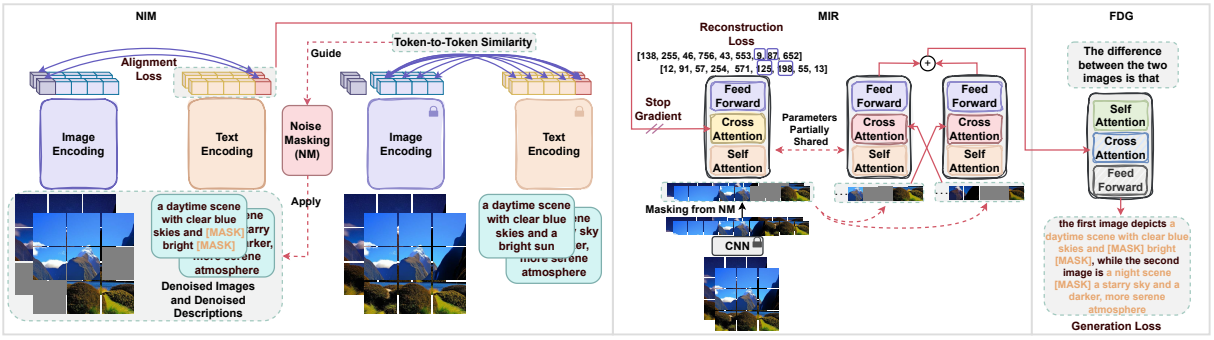



Figure 2: Illustration of the proposed noise-robust image difference captioning, including: Noise Identification and Masking (NIM), Masked Image Reconstruction (MIR), Fine-grained Description Generation (FDG).

*description]*" for each image in the pair. The subsequent NAMC model then leverages each image and its corresponding extracted description to compute their relevance, helping to identify and filter out irrelevant information in both modalities. To automate the generation of implicit change descriptions, we prompt QWen2-VL (Wang et al., 2024), which processes two input images, following the structured instruction (e.g., for CC3M):

> *Given the first and the second images, describe the difference between the two images. First image could be described as: {caption}. Second image could be described as: {caption}. Generate the description strictly according to the template: '{template}'.*

In the instruction, "*{caption}*" is a fine-grained paragraph detailing the image in CC3M, sourced from the existing LLava-ReCap dataset (Liu et al., 2024b). "*{template}*" is the structured template mentioned in Section 3.1.2. Further details on the generation of the fine-grained implicit-change descriptions are provided in Appendix D.

## 3.2 Noise-Aware Modeling and Captioning

### 3.2.1 Noise Identification and Masking

Given an image pair and its change description, our model feeds each image $I_i, i \in \{1, 2\}$ in the pair along with its corresponding description $T_i$, i.e. the text of "*[object description]*", into image and text encoders initialized with CLIP-B/16 (Radford et al., 2021). The dual encoders project data of both modalities into a shared embedding space.

**Image encoding.** Given an input image $I_i$, we first divide it into $n_I$ non-overlapping patches, which are then projected into patch representations $I_i^{pt} \in \mathbb{R}^{n_I \times d}$ using a convolutional layer, where $d$ is the hidden dimension. A learnable [CLS] token is prepended to the sequence, forming the combined input embedding. This embedding is processed by a 12-layer Transformer encoder $\mathcal{E}_I(\cdot)$, yielding the final image embedding $\mathcal{E}_I(I_i^{pt}) \in \mathbb{R}^{(n_I+1)\times d}$. Here, excluding the first token, the remaining forms the patch embedding $H_{I_i} = \mathcal{E}_I(I_i^{pt}) \in \mathbb{R}^{n_I \times d}$.

**Text encoding.** For the corresponding component text $T_i$, we prepend a start-of-sequence [SOS] token and append an end-of-sequence [EOS] token. The text is tokenized into $n_T$ subword units, which are then mapped to text embedding. The embedding is fed into a separate 12-layer Transformer encoder $\mathcal{E}_T(\cdot)$, producing the final text embedding $H_{T_i} = \mathcal{E}_T(T_i) \in \mathbb{R}^{n_T \times d}$.

**Token-to-token similarity.** The relevance between the image $I_i$ and description $T_i$ is quantified by the similarity between their embeddings. A token-to-token similarity matrix $S \in \mathbb{R}^{n_I \times n_T}$ from image to description is computed as:

$$S_{m,n} = \frac{H_{I_i,m}^T \cdot H_{T_i,n}}{||H_{I_i,m}|| \cdot ||H_{T_i,n}||}, \quad (1)$$

where $H_{I_i,m}$ and $H_{T_i,n}$ denote the $m$-th image patch embedding and $n$-th text token embedding, respectively, $S_{m,n}$ represents their cosine similarity, reflecting the token-to-token relevance between image and text. The elements in the relevance vector $R_{I_i} \in \mathbb{R}^{n_I}$ for the image $I_i$, is $R_{I_i,m} = \max(S_m)$, where $\max(S_m)$ denotes the maximum value in the $m$-th row of matrix $S$, representing the highest relevance between the $m$-th image patch and all text tokens. Symmetrically, the relevance vector $R_{T_i} \in \mathbb{R}^{n_T}$ for the text $T_i$, is computed by exchanging patch and text embeddings in Eq. (1).

**Noise identification and masking.** The noise identification for both modalities is achieved by measuring the irrelevance between them. Given the relevance vectors for image $R_{I_i}$ and text $R_{T_i}$, the corresponding irrelevance vectors are computed as $IR_{I_i} = \mathbf{1} - R_{I_i}$ and $IR_{T_i} = \mathbf{1} - R_{T_i}$, where $\mathbf{1}$ denotes a vector of ones with the same dimensions as the corresponding relevance vectors. On the basis, we propose Noise Masking

(NM), a strategy to mask irrelevant elements in each image and its corresponding textual description. Given the irrelevance vectors for both image $IR_{I_i}$ and text $IR_{T_i}$, tokens with lower relevance scores are assigned higher probabilities of being masked. Let $p_I$ be the masking probability ratio for image tokens and $p_T$ be the masking probability ratio for text tokens. The mask probability vectors for image and text, denoted as $\mathcal{P}_{I_i}$ and $\mathcal{P}_{T_i}$, are defined as $\mathcal{P}_{I_i} = p_I + IR_{I_i} - \mu(IR_{I_i})$ and $\mathcal{P}_{T_i} = p_T + IR_{T_i} - \mu(IR_{T_i})$, where $\mu(\cdot)$ denotes the mean operation.

**NIM output.** For each image $I_i$ in the pair and its description $T_i$, the NIM module's output is the respective image mask $M_{I_i} \in \mathbb{R}^{n_I}$ and the text mask $M_{T_i} \in \mathbb{R}^{n_T}$. These masks are derived from the corresponding probability vectors, $\mathcal{P}_{I_i}$ for image and $\mathcal{P}_{T_i}$ for text. Each element in $M_{I_i}$ is binary, where a value of 1 indicates the corresponding patch in the image $I_i$ is masked, and 0 indicates that it is preserved. The same binary interpretation applies to the elements in the text mask $M_{T_i}$.

**Denoised image.** During the masking process for the image $I_i$, the patch representation from $I_i^{pt}$ is replaced with a learnable vector $e^{msk} \in \mathbb{R}^d$, yielding an updated patch representation $I_i^{msk}$, referred to as the *denoised image*.

**Denoised description.** During the masking process for the description $T_i$, each word selected for masking is replaced with the word of [MASK], producing $T_i^{msk}$ as the *denoised description*. Furthermore, for each image pair, its corresponding *denoised implicit-change description* is created by combining the denoised descriptions of the first and second images, in the form as:

> *The difference between the two images is that the first image {descriptive verb} [denoised description], while the second image {descriptive verb} [denoised description].*

Here, "*[denoised description]*" is a placeholder for the respective denoised description.

### 3.2.2 Masked Image Reconstruction

Given the image $I_i$ and its mask $M_{I_i}$ produced by NM, an over-masking issue arises where positions of relevant patches tend to be mistakenly marked as masked in $M_{I_i}$. To address this, we introduce the MIR module, which comprises a masked image encoder trained to reconstruct the masked regions. The reconstruction process consists of four parts: (1) A pre-trained image tokenizer (Esser et al., 2021) converts each image $I_i$ in the pair into a discrete token sequence $z_i \in \mathbb{R}^{n_I}$, serving as the ground truth for reconstruction. (2) A pre-trained CNN (He et al., 2016) encodes the image $I_i$ into the raw image feature $X_i \in \mathbb{R}^{n_I \times C}$, where $C$ denotes the channel dimension. (3) For positions masked in the image mask $M_{I_i}$, the corresponding positions of feature elements in $X_i$ are replaced with a learnable embedding vector $e_M \in \mathbb{R}^C$, producing the masked image feature $X_i^M \in \mathbb{R}^{n_I \times C}$. (4) A masked image encoder processes $X_i^M$ and learns to reconstruct the positions where $e_m$ is inserted.

**Masked image embedding.** Given the masked image feature $X_i^M \in \mathbb{R}^{n_I \times C}$, a convolutional projection with a kernel size of one is applied to transform the features into a $d$-dimensional embedding. To encode spatial position information, a learnable positional embedding layer is added to the embedding, resulting in the updated embedding $\hat{X}_i^M \in \mathbb{R}^{n_I \times d}$.

**Masked image encoder.** This updated embedding is then fed into a two-layer Transformer for the reconstruction of regions replaced with $e_M$. Within the Transformer, the self-attention layer first processes $\hat{X}_i^M$, and the resulting output is used as the query in the subsequent cross-attention layer. The key and the value in the cross-attention layer, come from the text encoding of the denoised description $T_i^{msk}$ in NIM module, denoted as $\mathcal{E}_T(T_i^{msk}) \in \mathbb{R}^{n_T \times d}$. The specific reconstruction optimization is detailed in Section 3.3.

**MIR output.** The MIR module outputs the modeled cross-image interactions between the first and second images without attending to any noisy regions. To capture their interactions, we adopt a co-attentional style in the masked image encoder for the two input images, as shown in Figure 2. Specifically, given their masked image embedding, $\hat{X}_1^M$ and $\hat{X}_2^M$, the masked image encoder $\mathcal{E}_M$ processes them through two parallel streams. Each stream applies the self-attention layer to one image feature, producing an intermediate output that acts as the query in the subsequent cross-attention layer. The key and value of the cross-attention layer come from the other image feature, thereby enabling information exchange between both images. The outputs of the two streams are $\mathcal{E}_M(\hat{X}_1^M, \hat{X}_2^M) \in \mathbb{R}^{n_I \times d}$ and $\mathcal{E}_M(\hat{X}_2^M, \hat{X}_1^M) \in \mathbb{R}^{n_I \times d}$. These two outputs are then concatenated along the last dimension and passed through a fully connected layer to project the concatenated representation back to dimension $d$. This produces the final output of the MIR module, denoted as $\mathcal{E}_{MIR} \in \mathbb{R}^{n_I \times d}$. We en-

sure that masked regions in both images are not attended to by applying masking operations in both the self- and cross-attention layers during cross-image interaction modeling.

#### 3.2.3 Fine-grained Description Generation

Given the output $\mathcal{E}_{MIR}$ from the MIR module, the multi-modal decoder of a two-layer Transformer is optimized to predict the ground-truth of denoised implicit-change descriptions.

### 3.3 Optimization

**Optimization of NIM.** We optimize NIM's dual encoder using the contrastive loss proposed by Jiang and Ye (2023), to align cross-modal representations between each denoised image $I_i^{msk}$ in the pair and the corresponding denoised description $T_i^{msk}$. The NIM's dual encoders produce their embeddings of $\mathcal{E}_I(I_i^{msk})$ and $\mathcal{E}_T(T_i^{msk})$, respectively. To ensure that these embeddings are projected into a shared space, we extract their global semantic representations by selecting the [CLS] token embedding $e_{I_i}^{cls} \in \mathbb{R}^d$ from $\mathcal{E}_I(I_i^{msk})$ and the [EOS] token embedding $e_{T_i}^{eos} \in \mathbb{R}^d$ from the $\mathcal{E}_T(T_i^{msk})$. The alignment loss $\mathcal{L}_{\text{NIM}}$ for these embeddings is detailed in Appendix E.

**Optimization of MIR.** The MIR's masked image encoder is trained to reconstruct the masked regions in two images, guided by their respective denoised description. For each image, given the masked image embedding $\hat{X}_i^M$ and its respective denoised description $T_i^{msk}$, the masked image encoder produces the output $\mathcal{E}_M(\hat{X}_i^M, \mathcal{E}_T(T_i^{msk}))$. Following Bao et al. (2021), the reconstruction objective aims to predict masked regions to align with the corresponding values in the ground truth discrete token sequence $z_i$. To achieve this, the masked encoder output is fed through a fully connected layer and a softmax classifier to predict discrete tokens. The reconstruction objective $\mathcal{L}_{\text{MIR}}$ maximizes the log-likelihood of the correct tokens for both images in the pair, detailed in Appendix E.

**Optimization of FDG.** Leveraging the denoised implicit-change descriptions as textual ground truth, the FDG module's multi-modal decoder is trained to predict the next word by conditioning on previous words as well as the MIR's output, which captures cross-image interactions. Therefore, the generation loss $\mathcal{L}_{\text{FDG}}$ is to maximize the log-likelihood of the observed word sequences.

The overall loss $\mathcal{L}$ for optimizing our NAMC model is computed as the sum of the aforementioned losses from its three modules:

$$\mathcal{L} = \mathcal{L}_{\text{NIM}} + \mathcal{L}_{\text{MIR}} + \mathcal{L}_{\text{FDG}}. \quad (2)$$

| Model | B | M | C | R | S |
|---|---|---|---|---|---|
| **Streamlined Architecture** | | | | | |
| **ResNet Feataures** | | | | | |
| DUDA (2019) | 40.3 | 27.1 | 56.7 | – | 16.1 |
| VAM (2020) | 40.9 | 27.1 | 60.1 | – | 15.8 |
| VACC (2021) | 45.0 | 29.3 | 71.7 | – | 17.6 |
| MCCFormers (2021) | 46.9 | 31.7 | 71.6 | – | 14.6 |
| NCT (2023b) | 47.5 | 32.5 | 76.9 | 65.1 | 15.6 |
| VARD (2023a) | 48.3 | 32.4 | 77.6 | – | 15.4 |
| SCORER (2023c) | 49.4 | 33.4 | 83.7 | 66.1 | 16.2 |
| MURAT 2024 | 50.1 | 33.0 | 83.4 | 66.1 | 16.2 |
| NAMC (**Ours**) | **61.0** | **36.1** | **102.4** | **71.1** | **20.3** |
| **CLIP Features** | | | | | |
| CLIP4IDC (2022) | 54.7 | 33.0 | 89.9 | – | – |
| **LMM Parameter-efficient-finetuned Architecture** | | | | | |
| BLIP2IDC (2025) | 49.3 | 33.0 | 88.5 | – | – |

Table 1: Results on CLEVR-DC (CDC).

| Model | B | M | C | R | S |
|---|---|---|---|---|---|
| **Streamlined Architecture** | | | | | |
| **ResNet Features** | | | | | |
| VAM (2020) | 50.3 | 37.0 | 114.9 | 69.7 | 30.5 |
| VACC (2021) | 52.4 | 37.5 | 114.2 | – | 31.0 |
| $R^3$Net (2021a) | 54.7 | 39.8 | 123.0 | 73.1 | 32.6 |
| SGCC (2021) | 51.1 | 40.6 | 121.8 | 73.9 | 32.2 |
| SRDRL (2021b) | 54.9 | 40.2 | 122.2 | 73.3 | 32.9 |
| MCCFormers (2021) | 52.4 | 38.3 | 121.6 | – | 26.8 |
| BiDiff (2022) | 54.2 | 38.3 | 118.1 | – | 31.7 |
| PCL (2022) | 51.2 | 36.2 | 128.9 | 71.7 | – |
| NCT (2023b) | 55.1 | 40.2 | 124.1 | 73.8 | 32.9 |
| I3N-TD (2023) | 55.8 | 40.6 | 125.6 | 73.9 | 32.8 |
| VARD (2023a) | 55.4 | 40.1 | 126.4 | 73.8 | 33.3 |
| SCORER (2023c) | 56.3 | **41.2** | 126.8 | 74.5 | 33.3 |
| MURAT (2024) | 55.4 | 40.4 | 127.0 | 73.9 | 32.4 |
| SMART (2024b) | 56.1 | 40.8 | 127.0 | 74.2 | **33.4** |
| NAMC (**Ours**) | **57.5** | 38.6 | **153.2** | **77.3** | 27.6 |
| **CLIP Features** | | | | | |
| CLIP4IDC (2022) | 56.9 | 38.4 | 150.7 | 76.4 | – |
| **LMM Parameter-efficient-finetuned Architecture** | | | | | |
| VIR-VLFM (2023) | **58.2** | **42.6** | **153.4** | **78.9** | **34.5** |
| FINER (2024) | 55.6 | 36.6 | 137.2 | 72.5 | 26.4 |

Table 2: Results on CLEVR-Change (CLC).

## 4 Experiments

We pre-train our NAMC model and leverage the pre-trained MIR and FDG modules for on IDC fine-tuning. Notably, the NIM module is employed only during the pre-training. We evaluate our NAMC on the CLEVR-DC (CDC) (2021), CLEVR-Change (CLC) (2019), Image-Editing-Request (IER) (2019), and Spot-the-Diff (STD) (2018) datasets, with BLEU (B) (2002), METEOR (M) (2005), CIDEr (C) (2015), ROUGE-L (R) (2004), and Spice (2016).

**Baselines.** We compare the recent approaches that fall into two tracks: (1) Streamlined architecture based on either ResNet (2016) or CLIP (2021)

| Model | B | M | C | R | S |
|---|---|---|---|---|---|
| **Streamlined Architecture** | | | | | |
| **ResNet Features** | | | | | |
| MCCFormers (2021) | 8.3 | 14.3 | 30.2 | 39.2 | – |
| BiDiff (2022) | 6.9 | 14.6 | 27.7 | 38.5 | – |
| NCT (2023b) | 8.1 | 15.0 | 34.2 | 38.8 | 12.7 |
| VARD (2023a) | 10.0 | 14.8 | 35.7 | 39.0 | – |
| SCORER (2023c) | 10.0 | 15.0 | 33.4 | 39.6 | – |
| SMART (2024b) | 10.5 | 15.2 | 37.8 | 39.1 | – |
| NAMC (**Ours**) | **14.4** | **18.3** | **51.0** | **45.2** | **15.0** |
| **CLIP Features** | | | | | |
| CLIP4IDC (2022) | 8.2 | 14.6 | 32.2 | 40.4 | – |
| **LMM Parameter-efficient-finetuned Architecture** | | | | | |
| VIXEN-C (2024) | 8.6 | 15.4 | 38.1 | 42.5 | – |
| FINER (2024) | 14.1 | 15.9 | 53.3 | 40.4 | **15.9** |
| MGM+RP (2024) | 16.6 | 18.2 | 68.1 | 45.7 | – |
| BLIP2IDC (2025) | **17.4** | **20.1** | **74.1** | **48.5** | – |
| **LMM Fully-finetuned Architecture** | | | | | |
| OneDiff (2024) | 24.6 | 24.1 | 103.9 | 52.2 | – |

Table 3: Results on Image-Editing-Request (IER).

| Model | B | M | C | R | S |
|---|---|---|---|---|---|
| **Streamlined Architecture** | | | | | |
| **ResNet Features** | | | | | |
| VAM (2020) | 10.1 | 12.4 | 38.1 | 31.3 | – |
| VACC (2021) | 9.7 | 12.6 | 41.5 | 32.1 | – |
| $R^3$Net (2021a) | – | 13.1 | 36.6 | 32.6 | 18.8 |
| SRDRL (2021b) | – | 13.0 | 35.3 | – | 18.0 |
| MCCFormers (2021) | 10.0 | 12.4 | 43.1 | – | 18.3 |
| BiDiff (2022) | 6.6 | 10.6 | 42.2 | 29.5 | – |
| I3N-TD (2023) | 10.3 | 13.0 | 42.7 | 31.5 | 18.6 |
| VARD (2023a) | – | 12.5 | 30.3 | 29.3 | 17.3 |
| SCORER (2023c) | 10.2 | 12.2 | 38.9 | – | 18.4 |
| SMART (2024b) | – | **13.5** | 39.4 | 31.6 | **19.0** |
| MURAT (2024) | 10.2 | 13.1 | 39.4 | 33.1 | 18.8 |
| NAMC (**Ours**) | **11.8** | 13.4 | **52.3** | **33.4** | 18.7 |
| **CLIP Features** | | | | | |
| CLIP4IDC (2022) | 11.6 | 14.2 | 47.4 | 35.0 | – |
| **LMM Parameter-efficient-finetuned Architecture** | | | | | |
| VIR-VLFM (2023) | 12.2 | **15.3** | 48.9 | **36.2** | 20.1 |
| FINER (2024) | **12.9** | 14.7 | **61.8** | 35.5 | **22.1** |
| MGM+RP (2024) | 10.8 | 13.1 | 53.5 | 33.0 | – |
| BLIP2IDC (2025) | 11.4 | 13.5 | 51.4 | – | – |
| **LMM Fully-finetuned Architecture** | | | | | |
| OneDiff (2024) | 12.8 | 14.6 | 56.6 | 35.8 | – |

Table 4: Results on Spot-the-Diff (STD).

features. Specifically, we compare our NAMC with: DUDA (2019), VAM (2020), IFDC (2021), VACC (2021), $R^3$Net (2021a), SGCC (2021), SRDRL (2021b), MCCFormers (2021), BiDiff (2022), PCL (2022), CLIP4IDC (2022), NCT (2023b), I3N-TD (2023), VARD (2023a), SCORER (2023c), MURAT (2024), and SMART (2024b). (2) LMM-finetuned architectures, using parameter-efficient fine-tuning or fully fine-tuning. Specifically, we compare our NAMC with: VIR-VLFM (2023), VIXEN-C (2024), FINER (2024), MGM+RP (2024), OneDiff (2024), and BLIP2IDC (2025). Tables 1–4 present our results alongside state-of-the-art approaches across the four datasets. Our NAMC occupies a unique position outside both categories, yet combines the inference efficiency and performance.

| Model | #Total Params | Tokens Per Second (TPS) ↑ |
|---|---|---|
| CLIP4IDC (2022) | 173.7M | 1789.44 |
| FINER (2024) | 7.9B | 48.51 |
| BLIP2IDC (2025) | 3.8B | 106.17 |
| NAMC (**Ours**) | 64.2M | **2248.53** |

Table 5: Comparison on the numbers of parameters and efficiency during inference.

| Module | CLEVR-Change | | | | Spot-the-Diff | | | | Image-Editing-Request | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | B | M | C | R | B | M | C | R | B | M | C | R |
| NAMC | **57.5** | 38.6 | **153.2** | **77.3** | **11.8** | **13.4** | **52.3** | **33.4** | **14.4** | **18.3** | **51.0** | **45.2** |
| - MIR | 56.8 | 38.8 | 149.3 | 76.8 | 10.6 | 13.3 | 47.9 | 32.6 | 14.2 | 17.5 | 47.5 | 45.0 |
| - - NIM | 56.2 | **39.3** | 138.4 | 75.4 | 9.6 | 12.0 | 43.5 | 31.5 | 11.0 | 16.4 | 44.6 | 44.2 |

Table 6: Effects of combining different modules.

### 4.1 Results

When compared to streamlined architectures that also employ ResNet pre-extracted features, our NAMC consistently outperforms them across all datasets, demonstrating superior performance on the majority of metrics. We attribute this observation to NAMC's ability to address training data scarcity and mitigate noise interference during the pre-training stage. Compared to CLIP4IDC, which also uses a pre-trained model, our NAMC shows superior performance on all four datasets—for example, achieving improvements of 9.7% and 58.4% on the C score for STD and IER, respectively.

Compared with the state-of-the-art LMM-finetuned architectures, our NAMC shows competitive performance on the four datasets. For example, our NAMC model outperforms BLIP2IDC on the CDC, performs competitive with VIR-VLFM on the CLC, with FINER on the IER, and with MGM+RP on the STD, respectively. Table 5 further compares the inference efficiency of NAMC with a representative LMM-finetuned approach, FINER. Due to the architectural simplicity of our model, which incorporates a 2-layer Transformer for MIR and FDG modules, our NAMC demonstrates smaller numbers of parameters and more efficient inference, achieving a higher tokens-per-second (TPS) rate during inference. We provide detailed analyses of various components of our NAMC in Appendix A, and a comparison with more LMMs in Appendix C.

### 4.2 Ablation Studies

We conduct ablation studies on (1) modules, examining individual impacts, NM masking ratios, NIM pre-training alignment, and loss function choices; and (2) pre-training datasets, assessing different combinations and sizes. Additional ablation studies are provided in Appendix B.

| Loss | Spot-the-Diff | | | | Image-Editing-Request | | | |
|---|---|---|---|---|---|---|---|---|
| | B | M | C | R | B | M | C | R |
| **Different Contrastive Loss for NIM** | | | | | | | | |
| InfoNCE | 11.0 | 13.2 | 50.4 | 32.3 | 13.7 | 17.0 | 48.2 | 44.5 |
| Contrastive (2023) | **11.8** | **13.4** | **52.3** | **33.4** | **14.4** | **18.3** | **51.0** | **45.2** |
| **Different Construction Loss for MIR** | | | | | | | | |
| L1 | 9.6 | 12.6 | 40.6 | 31.1 | 10.7 | 16.3 | 41.0 | 43.5 |
| L2 | 9.7 | 12.7 | 40.4 | 30.5 | 10.2 | 16.8 | 40.8 | 43.0 |
| Discrete (2021) | **11.8** | **13.4** | **52.3** | **33.4** | **14.4** | **18.3** | **51.0** | **45.2** |

Table 7: Effects of different losses for NIM and MIR.

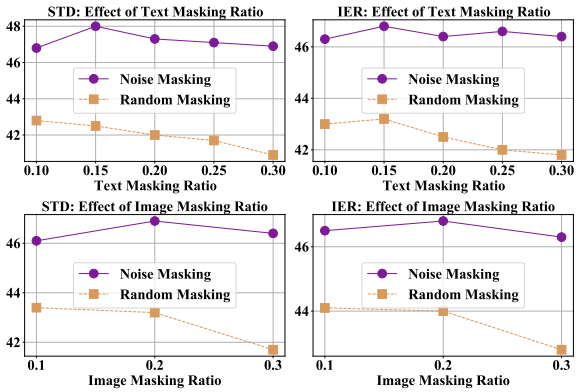



Figure 3: Effect of masking ratios within NM. We compare Noise Masking and Random Masking across different image and text masking ratios, $p_I$ and $p_T$. When varying $p_I$, we set $p_T = 0$, and vice versa.

### 4.2.1 Effect of Modules

**Effect of combining different modules.** Table 6 shows the combined impact of NIM module (for noise identification and masking) and MIR module (for image comprehension enhancement). Noise introduced in LMM-generated descriptions creates noisy image-text correspondences, degrading model performance. Incorporating NIM module during NAMC pre-training improves fine-tuning performance across most metrics on all datasets. Further integration of the MIR module, which mitigates over-masking issues and enhances image comprehension, brings the improvements. This demonstrates that integrating both modules consistently enhances fine-tuning performance across all three datasets, underscoring their effectiveness.

**Effect of different losses for NIM and MIR.** Within the NAMC model, we study exclusively on the selection of different losses for pre-training our NIM and MIR modules. Table 7 summarizes the selected loss functions: two contrastive losses for the NIM module and three construction losses for the MIR module. Our results indicate that, for the NIM module, the contrastive loss (2023) delivers superior performance, whereas for the MIR module, employing both L1 and L2 losses as a continuous reconstruction for the raw image feature $X_i$ causes critical details blurred among noisy information, thus degrading performance.

| Denoised Image | Denoised Desc. | Spot-the-Diff | | | | Image-Editing-Request | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | B | M | C | R | B | M | C | R |
| – | – | 10.2 | 12.5 | 46.1 | 32.4 | 12.6 | 15.9 | 46.1 | 43.4 |
| – | ✓ | 10.9 | 12.9 | 49.5 | 33.0 | 13.8 | 16.2 | 48.6 | 44.9 |
| ✓ | – | 10.6 | 13.0 | 48.7 | 32.8 | 13.9 | 16.5 | 48.2 | 44.5 |
| ✓ | ✓ | **11.8** | **13.4** | **52.3** | **33.4** | **14.4** | **18.3** | **51.0** | **45.2** |

Table 8: Effects of using denoised images and descriptions for NIM in pre-training.

**Effect of masking ratios within NM.** Figure 3 illustrates our study exclusively on various combinations of the masking ratios, $p_I$ and $p_T$, in the NM strategy. We compare NM with random masking, where all values in the mask probability vectors, $\mathcal{P}_{I_i}$ and $\mathcal{P}_{T_i}$, are set to $p_I$ and $p_T$, respectively. Generally, we observe a decline in model performance as the masking ratios increase in random masking. We attribute this to the increased loss of information caused by higher masking, which disrupts the model's performance. The dynamic masking in NM mitigates this issue. We select $p_I = 0.2$ and $p_T = 0.15$ as the masking ratios.

**Effect of aligning denoised images and descriptions for NIM in pre-training.** Table 8 presents our study exclusively on the effect of aligning denoised images and descriptions during pre-training the NIM module, i.e., aligning $\mathcal{E}_I(I_i^{msk})$ with $\mathcal{E}_T(T_i^{msk})$, as detailed in Section 3.3. We begin by aligning $\mathcal{E}_I(I_i^{pt})$ with $\mathcal{E}_T(T_i)$, then substitute either $\mathcal{E}_I(I_i^{pt})$ with $\mathcal{E}_I(I_i^{msk})$ or $\mathcal{E}_T(T_i)$ with $\mathcal{E}_T(T_i^{msk})$, achieving superior performance when employing both the denoised images or descriptions. This demonstrates that the NM strategy reduces noisy correspondences between image pairs and change descriptions, enhancing NAMC's performance.

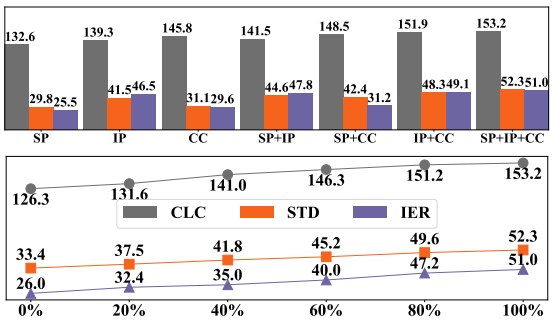
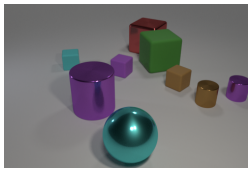
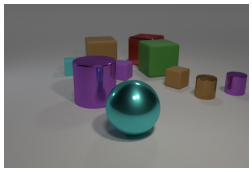



Figure 4: Effect of pre-training dataset combinations (top) and data sizes (bottom) on fine-tuning performance measured by the C Score. And SP, IP, and CC refer to image pairs from STD, InstructPix2Pix, and CC3M.

### 4.2.2 Effect of Pre-training Datasets

Figure 4 illustrates the impact of pre-training datasets combinations and sizes. We observe that the integration of datasets leads to improved performance on downstream tasks. Moreover, it is

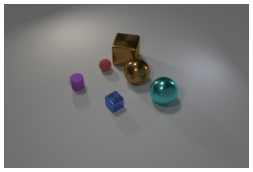
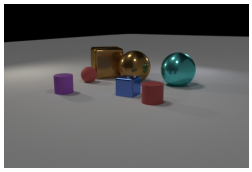



First image Second image

**Ground-truth:** the large brown matte block that is left of the red object has been newly placed
**Ours:** the big brown rubber block that is left of the big red metal thing has been newly placed

**FINER:** there is no change

(a) Comparison on CLC

First image Second image

**Ground-truth:** the other tiny object the same shape as the purple rubber object has been added
**Ours:** the other red cylinder that is the same size as the purple cylinder has been added

**BLIP2IDC:** the other purple object that is the same size as the purple cylinder changed its location

(b) Comparison on CDC


First image Second image

**Ground-truth:** people moved location
**Ours:** the people have moved

**FINER:** there are no people in the parking lot

(c) Comparison on STD


First image Second image

**Ground-truth:** increase the brightness of the entire image
**Ours:** brighten the entire photo

**FINER:** remove the cigarette

(d) Comparison on IER

Figure 5: Qualitative results on the four IDC datasets compared with FINER and BLIP2IDC. Incorrect and correct predictions are highlighted in red and green, respectively.

evident that the NAMC's performance steadily improves as the pre-training data amount increases. Additionally, we find that the gains on the CLC dataset exhibit smaller, expanding the training data from 80% to 100% yields a 1.3% improvement, compared to a 5.4% improvement on STD.

### 4.3 Qualitative Results

Figure 5 compares our NAMC with FINER on the CLC, STD, and IER datasets, and with BLIP2IDC on the CDC dataset. Our NAMC demonstrates strong robustness to background interference, effectively handling changes in viewpoint and maintaining focus on the content that undergoes actual change. We attribute this capability to the benefits of NAMC's pre-training strategy, which mitigates noisy cross-modal correspondences.

## 5 Conclusion

In this work, we focus on noise-robust image difference captioning at both the data and model levels. At the data level, we aim to leverage LMMs to generate fine-grained implicit-change descriptions for similar image pairs. At the model level, we propose NAMC: a Noise-Aware Modeling and Captioning model designed to mitigate cross-modal noisy correspondences introduced by the generated data. NAMC incorporates three trainable modules: a Noise Identification and Masking (NIM) module, a Masked Image Reconstruction (MIR) module, and a Fine-grained Description Generation (FDG) module. Experiments on four IDC benchmarks demonstrate the effectiveness of our NAMC.

## 6 Limitations

At the data level, we use an LMM to establish a data foundation for noise identification mechanisms at the model level, without relying on any pre-screening (Xu et al., 2025), knowledge injection (Li et al., 2025) or external noise detection (Jiao et al., 2024) pipelines. In the future, we will explore incorporating noise pre-screening pipelines to curate large-scale, high-quality data, leading to a accurate model that could produce coherent captions.

## Ethics Statement

This work involves the development and evaluation of a Noise-Aware Modeling and Captioning model (NAMC) for image difference captioning. The data for pre-training our NAMC is intelligently generated by QWen-2-VL, based on existing CC3M, InstructPix2Pix, and Spot-the-Diff datasets. The downstream datasets evaluated in our experiments contain CLEVR-Change, Spot-the-Diff, Image-Editing-Request, and CLEVR-DC. All these datasets and the QWen-2-VL are publicly available and have been granted a license to be used in the research community. We ensured that data usage complied with the respective licenses and terms of use. We acknowledge that artificial intelligence models may carry societal risks if deployed irresponsibly. These risks include potential misuse for surveillance, biased performance across demographic groups, and misinterpretation of visual content. While our work focuses on academic research, we strongly caution against the use of such models in high-stakes decision-making without careful fairness evaluations, transparency, and appropriate safeguards.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants 62476188, the Natural Science Foundation of the Jiangsu Higher Education Institutions of China, and the Academy of Finland (USSEE project, No. 345791). Computational resources are provided by the LUMI supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CSC and the LUMI consortium.

## References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 382–398. Springer.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. 2021. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*.

Djamila-Romaissa Beddiar, Mourad Oussalah, and Tapio Seppänen. 2023. Automatic captioning for medical imaging (mic): a rapid review of literature. *Artificial Intelligence Review*, 56(5):4019–4076.

Alexander Black, Jing Shi, Yifei Fan, Tu Bui, and John Collomosse. 2024. Vixen: Visual text comparison network for image difference captioning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 846–854.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. 2023. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402.

Shizhen Chang and Pedram Ghamisi. 2023. Changes to captions: An attentive network for remote sensing change captioning. *arXiv preprint arXiv:2304.01091*.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. 2024. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *Science China Information Sciences*, 67(12):220101.

Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883.

Gautier Evennou, Antoine Chaffin, Vivien Chappelier, and Ewa Kijak. 2025. Reframing image difference captioning with blip2idc and synthetic augmentation. *arXiv preprint arXiv:2412.15939*.

Zhongtian Fu, Kefei Song, Luping Zhou, and Yang Yang. 2024. Noise-aware image captioning with progressively exploring mismatched words. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 12091–12099.

Zixin Guo, Tzu-Jui Wang, and Jorma Laaksonen. 2022. Clip4idc: Clip for image difference captioning. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, pages 33–42.

Zixin Guo, Tzu-Jui Julius Wang, Selen Pehlivan, Abduljalil Radman, and Jorma Laaksonen. 2023. Pitl: Cross-modal retrieval with weakly-supervised vision-language pre-training via prompting. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2261–2265.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*.

Alain Hore and Djemel Ziou. 2010. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE.

Mehrdad Hosseinzadeh and Yang Wang. 2021. Image change captioning by learning from an auxiliary task. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2725–2734.

Genc Hoxha, Seloua Chouaf, Farid Melgani, and Youcef Smara. 2022. Change captioning: A new paradigm for multitemporal remote sensing image analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14.

Erdong Hu, Longteng Guo, Tongtian Yue, Zijia Zhao, Shuning Xue, and Jing Liu. 2024. Onediff: A generalist model for image difference captioning. In *Proceedings of the Asian Conference on Computer Vision*, pages 2439–2455.

Qingbao Huang, Yu Liang, Jielong Wei, Cai Yi, Hanyu Liang, Ho-fung Leung, and Qing Li. 2021. Image difference captioning with instance-level fine-grained feature representation. *IEEE Transactions on Multimedia*.

Zhenyu Huang, Mouxing Yang, Xinyan Xiao, Peng Hu, and Xi Peng. 2024. Noise-robust vision-language pre-training with positive-negative learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Harsh Jhamtani and Taylor Berg-Kirkpatrick. 2018. Learning to describe differences between pairs of similar images. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4024–4034.

Ding Jiang and Mang Ye. 2023. Cross-modal implicit relation reasoning and aligning for text-to-image person retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2787–2797.

Zhangqi Jiang, Junkai Chen, Beier Zhu, Tingjin Luo, Yankun Shen, and Xu Yang. 2025. Devils in middle layers of large vision-language models: Interpreting, detecting and mitigating object hallucinations via attention lens. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 25004–25014.

Qirui Jiao, Daoyuan Chen, Yilun Huang, Yaliang Li, and Ying Shen. 2024. Img-diff: Contrastive data synthesis for multimodal large language models. *arXiv preprint arXiv:2408.04594*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Jaakko Kainulainen, Zixin Guo, and Jorma Laaksonen. 2024. Diffusion-based multimodal video captioning. In *Proceedings of the Asian Conference on Computer Vision*, pages 2820–2837.

Wooyoung Kang, Jonghwan Mun, Sungjun Lee, and Byungseok Roh. 2023. Noise-aware learning from web-crawled image-text data for image captioning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2942–2952.

Hoeseong Kim, Jongseok Kim, Hyungseok Lee, Hyunsung Park, and Gunhee Kim. 2021. Agnostic change captioning with cycle consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2095–2104.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. 2024. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814*.

Yi Li, Yunbin Tu, Liang Li, Li Su, and Qingming Huang. 2025. Change entity-guided heterogeneous representation disentangling for change captioning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 17050–17060.

Zeming Liao, Qingbao Huang, Yu Liang, Mingyi Fu, Yi Cai, and Qing Li. 2021. Scene graph with 3d information for change captioning. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5074–5082.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Fenglin Liu, Changchang Yin, Xian Wu, Shen Ge, Ping Zhang, and Xu Sun. 2021. Contrastive attention for automatic chest x-ray report generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 269–280.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024b. Llava-next: Improved reasoning, ocr, and world knowledge.

Xiaonan Lu, Jianlong Yuan, Ruigang Niu, Yuan Hu, and Fan Wang. 2023. Viewpoint integration and registration with vision language foundation model for image change understanding. *arXiv preprint arXiv:2309.08585*.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Dong Huk Park, Trevor Darrell, and Anna Rohrbach. 2019. Robust change captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4624–4633.

Yingzhe Peng, Gongrui Zhang, Miaosen Zhang, Zhiyuan You, Jie Liu, Qipeng Zhu, Kai Yang, Xingzhong Xu, Xin Geng, and Xu Yang. 2025. Lmm-r1: Empowering 3b lmms with strong reasoning abilities through two-stage rule-based rl. *arXiv preprint arXiv:2503.07536*.

Yue Qiu, Shintaro Yamamoto, Kodai Nakashima, Ryota Suzuki, Kenji Iwata, Hirokatsu Kataoka, and Yutaka Satoh. 2021. Describing and localizing multiple changes with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1971–1980.

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

Abduljalil Radman and Jorma Laaksonen. 2025. Tsam: Temporal sam augmented with multimodal prompts for referring audio-visual segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 23947–23956.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565.

Xiangxi Shi, Xu Yang, Jiuxiang Gu, Shafiq Joty, and Jianfei Cai. 2020. Finding it at another side: A viewpoint-adapted matching encoder for change captioning. In *European Conference on Computer Vision*, pages 574–590. Springer.

Yaoqi Sun, Liang Li, Tingting Yao, Tongyv Lu, Bolun Zheng, Chenggang Yan, Hua Zhang, Yongjun Bao, Guiguang Ding, and Gregory Slabaugh. 2022. Bidirectional difference locating and semantic consistency reasoning for change captioning. *International Journal of Intelligent Systems*, 37(5):2969–2987.

Hao Tan, Franck Dernoncourt, Zhe Lin, Trung Bui, and Mohit Bansal. 2019. Expressing visual relationships via language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1873–1883.

Wentan Tan, Changxing Ding, Jiayu Jiang, Fei Wang, Yibing Zhan, and Dapeng Tao. 2024. Harnessing the power of mllms for transferable text-to-image person reid. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17127–17137.

Yunbin Tu, Liang Li, Li Su, Junping Du, Ke Lu, and Qingming Huang. 2023a. Adaptive representation disentanglement network for change captioning. *IEEE Transactions on Image Processing*.

Yunbin Tu, Liang Li, Li Su, Ke Lu, and Qingming Huang. 2023b. Neighborhood contrastive transformer for change captioning. *IEEE Transactions on Multimedia*.

Yunbin Tu, Liang Li, Li Su, Chenggang Yan, and Qingming Huang. 2024a. Distractors-immune representation learning with cross-modal contrastive regularization for change captioning. In *European Conference on Computer Vision*, pages 311–328. Springer.

Yunbin Tu, Liang Li, Li Su, Zheng-Jun Zha, and Qingming Huang. 2024b. Smart: Syntax-calibrated multi-aspect relation transformer for change captioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Yunbin Tu, Liang Li, Li Su, Zheng-Jun Zha, Chenggang Yan, and Qingming Huang. 2023c. Self-supervised cross-view representation reconstruction for change captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2805–2815.

Yunbin Tu, Liang Li, Chenggang Yan, Shengxiang Gao, and Zhengtao Yu. 2021a. R^3net: Relation-embedded representation reconstruction network for change captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9319–9329.

Yunbin Tu, Tingting Yao, Liang Li, Jiedong Lou, Shengxiang Gao, Zhengtao Yu, and Chenggang Yan. 2021b. Semantic relation-aware difference representation learning for change captioning. In *Findings of the association for computational linguistics: ACL-IJCNLP 2021*, pages 63–73.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Yingjia Xu, Mengxia Wu, Zixin Guo, Min Cao, Mang Ye, and Jorma Laaksonen. 2025. Efficient text-to-video retrieval via multi-modal multi-tagger derived pre-screening. *Visual Intelligence*, 3(1):1–13.

Linli Yao, Weiying Wang, and Qin Jin. 2022. Image difference captioning with pre-training and contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3108–3116.

Shengbin Yue, Yunbin Tu, Liang Li, Shengxiang Gao, and Zhengtao Yu. 2024. Multi-grained representation aggregating transformer with gating cycle for change captioning. *ACM Transactions on Multimedia Computing, Communications and Applications*.

Shengbin Yue, Yunbin Tu, Liang Li, Ying Yang, Shengxiang Gao, and Zhengtao Yu. 2023. I3n: Intra-and inter-representation interaction network for change captioning. *IEEE Transactions on Multimedia*.

Xian Zhang, Haokun Wen, Jianlong Wu, Pengda Qin, Hui Xue', and Liqiang Nie. 2024. Differential-perceptive and retrieval-augmented mllm for change captioning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 4148–4157.

# Appendix

## A Analyses on NAMC

In this section, we first analyze how our two proposed modules, the Noise Identification and Masking (NIM) module (Section 3.2.1) and the Masked Image Reconstruction (MIR) module (Section 3.2.2), function within our proposed NAMC pre-training. Then we analyze the robustness of our NAMC.

### A.1 NIM Analysis

We present qualitative results that demonstrate how our NIM module assigns relevance scores between images and their textual descriptions, conditioned by (1) without noisy contents, and (2) with noisy contents.

**Without noisy contents.** Figure 6 presents two examples, illustrating how the corresponding relevance scores are allocated. (1) For image: the regions identified as relevant are primarily those that exhibit a strong correspondence between the image areas and the associated text. For instance, in Figure 6(a), the highlighted regions focus on the player and the basketball, which are central to the textual description. In Figure 6(b), the regions mainly focus on the number and text written in the sand. (2) For text: when the same word appears multiple times in a sentence, its relevance scores vary depending on the context in which it occurs. The model tends to assign higher relevance to word instances that more directly depict the content of the image. For example, in the text of Figure 6(a), when several verbs are present, the relevance is concentrated on those that are most descriptive of the visual content. Specifically, the verb "catch" receives a higher score than "attempt to" in the first sentence, and "shot" is scored higher than both "attempt to" and "block". In the text of Figure 6(b), the relevance scores are assigned to the words "number," than "01" in the first sentence, and to "text" than "happy" in the second sentence. These examples show the aligned relevance between the image and the corresponding text provided by our NIM module.

**With noisy contents.** Figures 7 and 8 further present three examples. We mainly focus on the relevance score on text. (1) Figure 7(a): The relevance is primarily focused on the background elements, such as the wooden surface and the plain white area, while the noisy word "bee" receives a relatively low score. (2) Figure 7(b): In the second sentence, noisy content exists—specifically, the word "ball," which does not appear in the image, is assigned a low relevance score. Additionally, phrases like "impact or just before it" are also considered irrelevant and receive low scores. (3) Figure 8: The first sentence contains a large amount of noisy content, and our NIM assigns low relevance scores to most of its words. Similarly, in the first image, when the described object is irrelevant, the relevance scores are correspondingly distributed over the background.

**Overall.** Our NIM module effectively mitigates cross-modal noisy correspondences introduced by the generated data. However, it inevitably introduces over-masking, where relevant information is mistakenly masked by the NIM module.

First image Second image

**Generation:** The difference between two images is that the first image captures a player in a green and yellow uniform attempting to catch the basketball, while the second image shows a player in a blue jersey attempting to block the shot.

First image Second image

**Generation:** The difference between two images is that the first image features the number “2017” written in the sand, while the second image contains the text “happy new year” written in the sand.


(a) First image pair.


(b) Second image pair.

Figure 6: Relevance scores for two examples, without noisy contents. Relevant image regions are highlighted in blue, and a color bar indicates textual relevance scores.

## A.2 MIR Analysis

To mitigate the risk of over-masking for image, we introduce the MIR module. By reconstructing missing image content through a masked image encoder, the MIR module learns robust feature representations that preserve critical contextual details despite masked patches. To evaluate MIR’s ability to recover masked image patches, we conduct a quantitative analysis on MIR and its relationship with the VQGAN image tokenizer (Esser et al., 2021) that guides MIR process (see Section 3.2.2). Using a set of 1000 unseen image pairs from InstructPix2Pix (Brooks et al., 2023), we measure patch quality and similarity using PSNR (Hore and Ziou,

First image

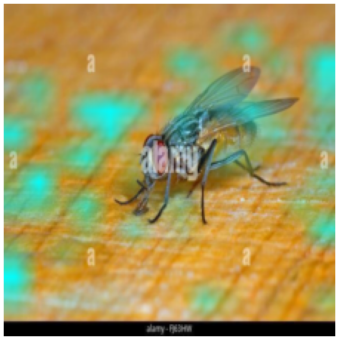
Second image

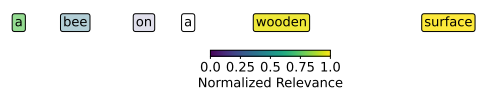
**Generation:** The difference between two images is that the first image shows a bee on a wooden surface, while the second image shows a bee against a plain white background.

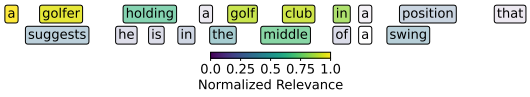
First image

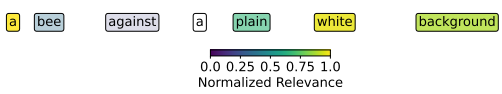
Second image

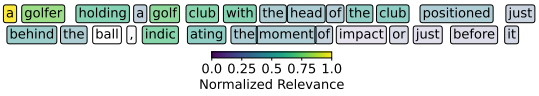
**Generation:** The difference between two images is that the first image features a golfer holding a golf club in a position that suggests he is in the middle of a swing, while the second image shows a golfer holding a golf club with the head of the club positioned just behind the ball, indicating the moment of impact or just before it.






(a) First image pair. (b) Second image pair.

Figure 7: Relevance scores for two examples, with noisy contents highlighted as the red color.

2010) and SSIM (Hore and Ziou, 2010). Table 9 summarizes these results, presenting comparisons:

**(1) MIR vs. VQGAN**: The first row indicates a strong similarity between the reconstructed patches from our MIR module and the patches from the VQGAN image tokenizer. With an SSIM of 0.805 and a PSNR of 27.74, these metrics are the highest in the table, suggesting that our MIR effectively generates content highly consistent with the VQ-GAN's output.

**(2) VQGAN vs. Raw and MIR vs. Raw**: The second and third rows reveal a notable decrease in

First image Second image

**Generation:** The difference between two images is that the first image shows a person jumping, with each frame capturing the person in mid-air, while the second image depicts a bald eagle standing on the edge of a body of water, looking downwards.

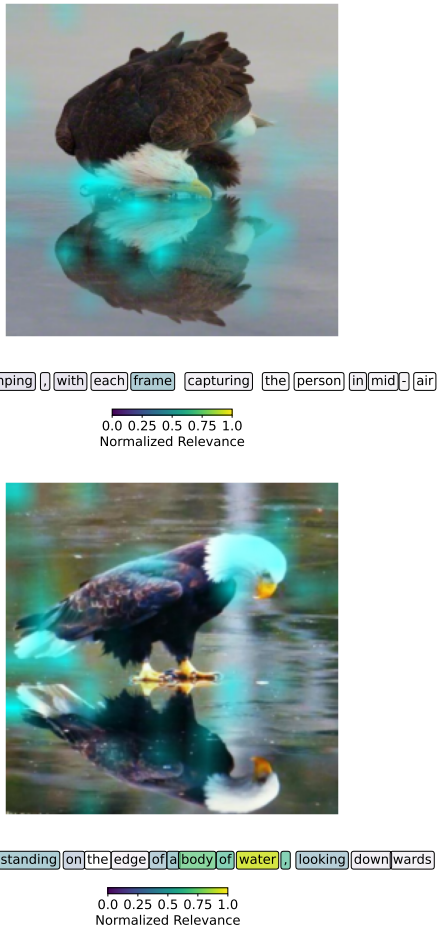



Figure 8: Relevance scores for an example, with a lot of noisy contents.

similarity when comparing either VQGAN patches (PSNR 25.89, SSIM 0.710) or MIR patches (PSNR 23.35, SSIM 0.616) to the original raw image patches. This highlights that both VQGAN and our MIR representation have information loss from the raw image data.

**(3) Overall Relationship**: The results demonstrate that MIR is highly similar to VQGAN (first row), yet both show reduced similarity to the raw image (third and second rows). The observed dissimilarity between MIR and the raw image is primarily attributed to the information loss inherent in the VQGAN transformation from raw data, which is reflected in the MIR patches because of MIR's strong correspondence with the VQGAN output.

| Reconstructed Patch | Reference Patch | PSNR↑ | SSIM↑ |
|---|---|---|---|
| MIR patch | VQGAN patch | 27.74 | 0.805 |
| VQGAN patch | Raw image patch | 25.89 | 0.710 |
| MIR patch | Raw image patch | 23.35 | 0.616 |

Table 9: Quantitative results on reconstructed patch quality and similarity of MIR and VQGAN.

### A.3 Robustness Analysis

We present an empirical evaluation of our NAMC in Table 10, focusing on captioning robustness under various noisy conditions. Specifically, we fine-tune the pre-trained NAMC on the downstream IER dataset, which features distribution shifts in both text and image modalities. For text, we introduce four levels of word-replacement noise—affecting 20%, 40%, 60%, and 80% of tokens—while for images we apply two corruption types: Gaussian noise and Fog. The image corruptions are with severity level of 5, defined by Hendrycks and Dietterich (2019). Our results demonstrate that NAMC maintains robust performance under these shift distributions across modalities.

| Shift Type | NAMC | B | M | C | R |
|---|---|---|---|---|---|
| Clean | w/o NIM and MIR | 11.0 | 16.4 | 44.6 | 44.2 |
| | w/ all | **14.4** | **18.3** | **51.0** | **45.2** |
| Replacing 20% Word | w/o NIM and MIR | 9.7 | 14.6 | 35.8 | 41.2 |
| | w/ all | **11.7** | **15.9** | **43.4** | **43.1** |
| Replacing 40% Word | w/o NIM and MIR | 8.4 | 13.7 | 33.5 | 39.7 |
| | w/ all | **10.7** | **14.7** | **39.0** | **41.9** |
| Replacing 60% Word | w/o NIM and MIR | 4.4 | 9.6 | 21.0 | 33.5 |
| | w/ all | **5.8** | **10.8** | **23.7** | **35.8** |
| Replacing 80% Word | w/o NIM and MIR | 0.8 | 3.5 | 2.1 | 9.9 |
| | w/ all | **0.9** | **3.7** | **2.3** | **10.2** |
| Gaussian Noise | w/o NIM and MIR | 10.1 | 15.2 | 34.9 | 41.6 |
| | w/ all | **12.1** | **16.7** | **43.1** | **43.6** |
| Fog Weather | w/o NIM and MIR | 10.3 | 14.9 | 34.5 | 40.8 |
| | w/ all | **11.8** | **15.4** | **42.2** | **41.3** |

Table 10: Evaluation of NAMC's robustness under various noise shift types.

## B Additional Ablation Studies

We provide the study on (1) architectures, including the effect of the number of layers our MIR and FDG

modules; (2) data generation approaches using different prompts for the LMM; and (3) sensitivity to the order of the first and second image–caption pairs in NAMC pre-training.

### B.1 Effect of Architectures

Figure 9 illustrates the impact of the number of layers in the MIR and FDG modules. We select the model with 2 hidden layers for both MIR's encoder and FDG's decoder, as it shows a good balance between performance and computation.

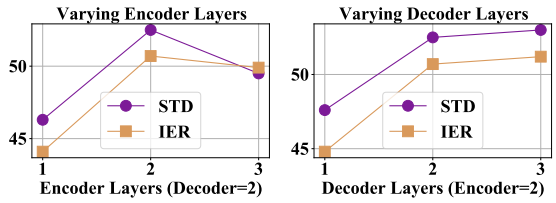



Figure 9: Effect of the number of layers on the MIR's masked image encoder and FDG's multi-modal decoder.

### B.2 Effect of Data Generation Approaches

We compare our data generation approach, presented in Section 3.1, with a conventional approach for change caption generation. This approach prompts the LMM, Qwen2-VL-7B-Instruct (Wang et al., 2024), to generate the description without following any templates, given the prompt:

> *Given the first and the second images, describe the change between the two images. What has changed from the first image to the second?*

Table 11 shows their comparison on NAMC without NIM and MIR on the downstream STD and IER datasets. It indicates that our change caption generation further gives quantitative improvements compared with the conventional caption generation. When prompting Qwen2-VL-7B-Instruct with the conventional approach, it tends to generate descriptions as short sentences, which loses some granularity compared to the output of our data generation approach, as shown in an example in Figure 10.

| Approach | Model | Spot-the-Diff | | | | Image-Editing-Request | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | B | M | C | R | B | M | C | R |
| Conventional | NAMC without | 9.4 | **12.5** | 41.7 | 30.1 | **11.3** | 16.0 | 42.3 | 42.8 |
| Ours | NIM and MIR | **9.6** | 12.0 | **43.5** | **31.5** | 11.0 | **16.4** | **44.6** | **44.2** |

Table 11: Effects of different data generation approaches for our NAMC without NIM and MIR.

### B.3 Effect of Image Orders in Pretraining

We have included a sensitivity analysis by randomly (50% probability of swapping the first and second) exchange both the order of images and their corresponding captions during NAMC's pre-training stage. As shown in Table 12, we compare NAMC pre-training with and without order exchange on the downstream IER dataset. The results indicate that NAMC is not sensitive to the order of the two input images.

First image


Second image

**Conventional:** The word hunger is no longer there. the keys are different.

**Ours:** The difference between two images is that the first image shows a typewriter with the word "HUNGER" typed on the paper, while the second image shows a typewriter with the keys visible and no text on the paper.

Figure 10: An example of the comparison between ours and conventional data generation approaches.

| Setting | B | M | C | R |
|---|---|---|---|---|
| With order exchange | **14.6** | 17.3 | 50.5 | **45.5** |
| Without order exchange | 14.4 | **18.3** | **51.0** | 45.2 |

Table 12: Sensitivity to the order of the first and second image-caption pairs in NAMC pre-training.

## C More Comparisons with LMMs

Table 13 provides a further comparison between our NAMC and four LMMs that are not fine-tuned, QWen2-VL-7B-Instruct (2024), LLaVA-1.5-7B (2024a), MGM-7B (2024), InternVL2-8B-FT (2024), on STD and IER datasets. It shows that QWen2-VL-7B-Instruct performs best among these

four LMMs, evidenced by its superior performance across most of the metrics. While our NAMC outperforms state-of-the-art LMMs across all metrics on STD. On IER, we achieve competitive results to LLaVA and outperform InternVL2-8B-FT in three metrics, with superior performance in metric M compared to all models.

| Model | Spot-the-Diff | | | | Image-Editing-Request | | | |
|---|---|---|---|---|---|---|---|---|
| | B | M | C | R | B | M | C | R |
| NAMC (Ours) | **11.8** | **13.4** | **52.3** | **33.4** | 14.4 | **18.3** | 51.0 | 45.2 |
| Qwen2-VL-7B-Instruct (2024) | 7.5 | 13.0 | 47.5 | 32.3 | **19.4** | 17.4 | **68.4** | **45.5** |
| LLaVA-1.5-7B (2024a) | 8.5 | 12.0 | 38.3 | 30.1 | 15.1 | 17.8 | 60.6 | 45.2 |
| MGM-7B (2024) | 9.9 | 12.0 | 46.3 | 31.5 | 16.5 | 17.7 | 66.8 | 44.8 |
| InternVL2-8B-FT (2024) | 6.6 | 11.7 | 26.5 | 27.3 | 12.4 | 14.1 | 51.5 | 38.9 |

Table 13: Comparison of our NAMC with four LMMs on STD and IER datasets.

# D More Details on Data Generation

## D.1 Similar Image Pairs

In this work, we collect an image dataset comprising approximately 770K synthetic and real image pairs. We use the existing dataset provided by InstructPix2Pix (Brooks et al., 2023) to get 313K synthetic image pairs. For real image pairs, we leverage two sources: 8K image pairs are from the existing Spot-the-Diff dataset (Jhamtani and Berg-Kirkpatrick, 2018), and 450K image pairs are collected from the CC3M dataset (Sharma et al., 2018). Unlike the datasets provided by Brooks et al. (2023) and Jhamtani and Berg-Kirkpatrick (2018), directly collecting similar image pairs from CC3M is not feasible, as the images in this dataset are not grouped together. In this section, we will describe the methods used to collect similar image pairs from the CC3M dataset.

### D.1.1 Task Overview

Similar image pairs can be defined as images that share similar semantic context (not identical), such as background, objects, textures, events, and other related elements. Given a set of images $D = \{I_1, ..., I_N\}$, our goal is to construct a set of similar image pairs $S = \{(I_i, I_j)\}$, where $i, j \in \{1, ..., N\}$, and $i \neq j$. To assess the similarity between images, we extract features using off-the-shelf image encoders, CLIP (Radford et al., 2021) and DINOv2 (Oquab et al., 2023), and then compute the similarity between these image features to build candidate image sets. Then, we propose a cluster-based approach to construct the final set of similar image pairs.

### D.1.2 Candidate Set Construction

For each image $I_i \in D$, we first extract its CLIP and DINOv2 features as follows:

$$f_i^{\text{CLIP}} = \mathcal{E}_{\text{CLIP}}(I_i), \; f_i^{\text{DINO}} = \mathcal{E}_{\text{DINO}}(I_i), \quad (3)$$

where $\mathcal{E}_{\text{CLIP}}$ and $\mathcal{E}_{\text{DINO}}$ denote the image encoder of CLIP and DINOv2, respectively. Next, we compute the similarity between each image based on its CLIP and DINOv2 features:

$$s^\theta = \text{sim}(f_i^\theta, f_j^\theta) \quad (4)$$

where $i, j \in \{1, ..., N\}$, and $\theta$ indicates whether the feature is extracted using CLIP or DINOv2. With the help of Faiss (Johnson et al., 2019), we efficiently compute and retrieve the top $k$ most similar images to $I_i$ based on both CLIP and DINOv2 features, resulting in two candidate image sets for $I_i$:

$$C_i^{\text{CLIP}} = \{I_1^{\text{CLIP}}, ..., I_k^{\text{CLIP}}\}, \quad (5)$$

$$C_i^{\text{DINO}} = \{I_1^{\text{DINO}}, ..., I_k^{\text{DINO}}\}. \quad (6)$$

To build the target candidate set from $C_i^{\text{CLIP}}$ and $C_i^{\text{DINO}}$, we apply thresholds $\{\tau_{max}, \tau_{min}\}$ to retain images whose similarity is neither too high (indicating identical images) nor too low (indicating dissimilar images) within each candidate set. We also apply another set of thresholds $\{\tau_{max}^{\text{CLIP}}, \tau_{min}^{\text{CLIP}}, \tau_{max}^{\text{DINO}}, \tau_{min}^{\text{DINO}}\}$ to ensure that the candidate set falls within an appropriate range by filtering out any sets where the maximum similarity is below $\tau_{max}^\theta$ or minimum similarity is below $\tau_{min}^\theta$. The final candidate image set is constructed as $C_i = \{I_c\}$, where $I_c$ denotes the images that appear in both $C_i^{\text{CLIP}}$ and $C_i^{\text{DINO}}$ after filtering.

### D.1.3 Cluster-based Similar Image Pair Construction

Directly constructing similar image pairs between $I_i$ and every image in $C_i$ is suboptimal due to potential redundancy, as $C_i$ may contain numerous semantically similar images. To mitigate this issue, we propose grouping the images in $C_i$ into categories, where intra-class similarities and inter-class variations are maximized. Then we construct image pairs according to each group. Thus, we introduce a cluster-based approach to generate the final set of similar image pairs.

For $I_c \in C_i \cup \{I_i\}$, we concatenate its CLIP and DINOv2 features, as computed in Eq. (3), to

form a target feature for computing the similarity in clustering:

$$f_c = g(I_c), \tag{7}$$

$$g(I) = \text{concat}(\alpha \mathcal{E}_{\text{CLIP}}(I), \beta \mathcal{E}_{\text{DINO}}(I)), \tag{8}$$

where $\text{concat}(\cdot, \cdot)$ denotes the concatenation operation, and $\alpha$ and $\beta$ are hyperparameters used to weight the two features, as the CLIP feature and DINOv2 feature have different scales.

Then we cluster $C_i$ into $K$ clusters with the features computed in Eq. (7):

$$\{C_{i,1}, ..., C_{i,K}\}. \tag{9}$$

For each cluster, we select the image most similar to $I_i$ as the target image:

$$I^*_{i,k} = \underset{I_{i,t} \in C_{i,k}}{\text{argmax}} \ \text{sim}(g(I_i), g(I_{i,t})), \tag{10}$$

where $k \in \{1, ..., K\}$.

Finally, we can get the similar image pairs of $I_i$ as follows:

$$S_i = \{(I_i, I^*_{i,1}), ..., (I_i, I^*_{i,K})\}. \tag{11}$$

The total set of similar image pairs is then given by:

$$S = \bigcup_{i \in \{1,...,N\}} S_i. \tag{12}$$

#### D.1.4 Implementation Details

We use ViT-L-14 as the CLIP image encoder and a large backbone size for the DINOv2 image encoder. The thresholds $\{\tau_{max}, \tau_{min}\}$ are set to $\{0.95, 0.78\}$, and the thresholds $\{\tau^{\text{CLIP}}_{max}, \tau^{\text{CLIP}}_{min}, \tau^{\text{DINO}}_{max}, \tau^{\text{DINO}}_{min}\}$ are set to $\{0.80, 0.78, 0.80, 0.78\}$. Spectral clustering is employed to group the candidate images into 3 clusters ($K = 3$), utilizing a radial basis function (RBF) kernel with a bandwidth parameter $\gamma = 0.1$ to construct the similarity matrix. The hyperparameters $\alpha$ and $\beta$ are set to 0.4 and 0.6 respectively.

### D.2 Difference Descriptions

For image pairs from Spot-the-Diff, InstructPix2Pix, and CC3M, we prompt LMMs with structured instructions to generate a single difference description for each pair. Specifically, for image pairs from Spot-the-Diff, the structured instruction is as:

> *Given the first and the second images, describe the difference between the two images. First image could be described as: {caption}. Second image could be described as: {caption}. Here is the difference: {difference}. Generate the description strictly according to the template: '{template}'.*

Here, "*{difference}*" is a summarized change description provided by Spot-the-Diff. For image pairs from the InstructPix2Pix, the structured instruction is as:

> *Given the first image and a prompt, the second image is generated based on them. Describe the difference between the two images. Here is the prompt: {instruction}. First image could be described as: {caption}. Second image could be described as: {caption}. Generate the description strictly according to the template: '{template}'.*

Here, "*{prompt}*" is a summarized instruction for how to edit the image, provided by the InstructPix2Pix. For image pairs from the CC3M, the structured instruction is as:

> *Given the first and the second images, describe the difference between the two images. First image could be described as: {caption}. Second image could be described as: {caption}. Generate the description strictly according to the template: '{template}'.*

Overall, "*{caption}*" is a fine-grained paragraph detailing the image in the three datasets, sourced from the existing LLava-ReCap dataset (Liu et al., 2024b). For each image in Spot-the-Diff and InstructPix2Pix, we use LLava to generate the fine-grained paragraph. The template is shown as:

> *The difference between the two images is that the first image {descriptive verb} [object description], while the second image {descriptive verb} [object description].*

In the template, "*{descriptive verb}*" is a placeholder for a replaceable verb chosen from a predefined list:

> *{is, describes, depicts, illustrates, , displays, demonstrates, introduces, has, positions, starts, places, discusses, promotes, takes, converts, adopts, presents, reveals, incorporates, aims, refers, offers, focuses, suggests, attributes, specifies, captures, symbolizes, advertises, conveys, features, appears, consists, remains, shows, lists, explains, shifts, indicates, provides, contains, exhibits, labels, sets, looks, states, showcases, creates, centers}.*

During the description generation process, we

observe cases that the LMM we use, Qwen2-VL-7B-Instruct (Wang et al., 2024), generates sentences that do not follow our template. During the description extraction process, we then filter out with automated paradigm those not following the given template, or containing only negations and auxiliary verbs, such as "something does/is/was/was not/is not/does not". Overall, about 0.2% of the descriptions are rejected in total.

### D.3 Generated Descriptions Diversity

We show the statistics of repetitions in our generated descriptions in Table 14. We do not observe a strong degree of sentence repetition, 3%, in the 770K generated sentences. We find that the SP dataset, which is generated from downstream STD dataset—a domain-specific dataset—exhibits a relatively higher repetition in generations compared to IP and CC.

| Dataset | Total | Unique | Repetition | Repetition Ratio |
|---|---|---|---|---|
| IP | 311,499 | 297,485 | 14,014 | 4.49% |
| CC | 449,194 | 440,204 | 8,990 | 2.00% |
| SP | 7,004 | 6,439 | 565 | 8.06% |
| IP+CC+SP | 767,697 | 744,128 | 23,569 | 3.07% |

Table 14: The statistics of the generated descriptions for NAMC pre-training.

## E Optimization of NIM and MIR

We present more details about the optimization of our NIM and MIR modules.

### E.1 Optimization of NIM

During training, each mini-batch contains two subsets: (1) $B$ pairs of masked first images and their corresponding masked component texts, and (2) $B$ pairs of masked second images and their masked component texts. To formalize the contrastive alignment objective, we illustrate the procedure using the first subset; the second subset follows a similar formulation. For the global representation of any image in the first subset, $e^{cls}_{I_i}$, a contrastive alignment set is constructed as $\{(e^{cls}_{I_i}, e^{eos}_{T_j}), y_{i,j}\}^{B}_{j=1}$, where $y_{i,j} = 1$ if the image $I_i$ is paired with the component text $T_j$, and $y_{i,j} = 0$ otherwise. The ground-truth matching distribution, $q_i$, is then defined as:

$$q_{i,j} = y_{i,j}/|y_i|, |y_i| = \sum_{j=1}^{B} y_{i,j}. \tag{13}$$

While the modeled matching distribution, $p_i$, parameterized by the similarity scores between image-text pairs, is defined as:

$$p_{i,j} = \frac{\exp(\text{sim}(e^{cls}_{I_i}, e^{eos}_{T_j})/\tau)}{\sum_{k=1}^{B} \exp(\text{sim}(e^{cls}_{I_i}, e^{eos}_{T_k})/\tau)}, \tag{14}$$

where $\text{sim}(\cdot,\cdot)$ computes the cosine similarity, and $\tau$ is a temperature hyperparameter. For the first subset in a mini-batch, the image-to-text alignment loss is defined as: $\mathcal{L}_{i2t} = KL(p_i||q_i)$, where $KL(\cdot||\cdot)$ denotes the Kullback-Leibler divergence between the modeled distribution $p_i$ and the ground-truth matching distribution $q_i$. Similarly, the text-to-image alignment loss, $\mathcal{L}_{t2i}$, is formulated by exchanging the roles of $e^{cls}_{I_i}$ and $e^{eos}_{T_j}$ in Eq. (14). The total alignment loss for the first subset is:

$$\mathcal{L}_{I_1} = \mathcal{L}_{i2t} + \mathcal{L}_{t2i}. \tag{15}$$

In summary, the total alignment loss across the mini-batch is computed as the sum of such losses from both subsets as:

$$\mathcal{L}_{\text{NIM}} = \mathcal{L}_{I_1} + \mathcal{L}_{I_2}. \tag{16}$$

### E.2 Optimization of MIR

The MIR's masked image encoder processes the masked image $\hat{X}^M_i$ and relevant textual information (e.g., $T^{msk}_i$ processed by a text encoder $\mathcal{E}_T$) to produce output features $\hat{Y}_i = \mathcal{E}_M(\hat{X}^M_i, \mathcal{E}_T(T^{msk}_i)) \in \mathbb{R}^{n_I \times d}$. For each position $j \in \{1, \ldots, n_I\}$, the corresponding feature vector $\hat{Y}_{ij} \in \mathbb{R}^d$ is projected by a fully-connected layer and then a softmax function. This yields a probability distribution over a vocabulary of $V$ discrete image tokens. We denote the predicted probability of the token at position $j$ being $v$ (where $v \in \{1, \ldots, V\}$) as:

$$p(\hat{z}_j = v|\hat{Y}_{ij}), \tag{17}$$

where $\hat{z}_j$ is the random variable for the token at position $j$.

Let $z_i = (z_{i1}, z_{i2}, \ldots, z_{in_I})$ be the ground-truth sequence of token indices for the $n_I$ target positions corresponding to image $I_i$, where each $z_{ij} \in \{1, \ldots, V\}$. The final objective of MIR is defined by the negative log-likelihood loss (cross entropy) over the target tokens:

$$\mathcal{L}_{\text{MIR}} = -\mathbb{E}_{z_i \sim \mathcal{D}} \left[ \sum_{j=1}^{n_I} \log p(\hat{z}_j = z_{ij}|\hat{Y}_{ij}) \right]. \tag{18}$$

Here, $p(\hat{z}_j = z_{ij}|\hat{Y}_{ij})$ is the probability assigned by the model to the ground-truth token $z_{ij}$ at the $j$-th position, given the encoder feature $\hat{Y}_{ij}$. The expectation $\mathbb{E}$ is typically taken over the training data distribution $\mathcal{D}$. In practice, we compute this loss only over the masked positions (i.e. those j for which $M_{I_i,j} = 1$).

## F Implementation Details

We describe more details of pre-training our NAMC, and fine-tuning our model as follows.

**Pre-training.** We initialize the dual encoder in the NIM module with CLIP-ViT-B/16 (Radford et al., 2021). For the MIR module, we employ a pre-trained ResNet-101 (He et al., 2016) to extract the grid features of each image, with dimensions of $14 \times 14$ and 1024 channels. Following this, the image features of the two images are reduced to a dimension of 768 and then fed into the masked image encoder in the MIR module, which comprises two standard Transformer layers (Vaswani et al., 2017), each with 12 attention heads and a hidden dimension of 768. For generating the change descriptions between the two images, we employ a multi-modal decoder with two standard Transformer layers, each comprising 12 attention heads and a hidden dimension of 768. Pre-training is carried out for 20 epochs using the Adam optimizer, with an initial learning rate of 0.0001 and a minimum learning rate of 0.00001 for the NIM module. For the MIR and FDG modules, the Adam optimizer is employed with an initial learning rate of 0.0003 and a minimum learning rate of 0.0001. The training is performed with a batch size of 512, distributed across 8 AMD MI250x GPUs.

**Fine-tuning.** We fine-tune the MIR and FDG modules in NAMC model on three benchmark IDC tasks. The fine-tuning is performed for 40 epochs on CLEVR-Change and CLEVR-DC, 10 epochs on Spot-the-Diff, and 20 epochs on Image-Editing-Request datasets, all with an initial learning rate of 0.0002 and a batch size of 128. During inference, the model generates sequences with a maximum length of 23 tokens using greedy search. Both pre-training and fine-tuning are implemented using PyTorch, and the results are computed utilizing the Microsoft COCO evaluation package. The fine-tuning is performed on 1 AMD MI250x GPU. The evaluation of model efficiency during inference, as shown in Table 5, is conducted with one RTX3090 on the IER dataset.