

OpenIDC: Reconstructing the Evidence Chain for Lightweight Image Difference Captioning

Jiabao Du¹, Saihui Hou^{1,*}, Yuan Liu¹, Jiawei Wang², Yongzhen Huang^{1,3}

¹Beijing Normal University, Beijing, China

²Beijing Jiaotong University, Beijing, China ³WATRIX.AI, Beijing, China

{dujiabao2025, liuyuan2024}@mail.bnu.edu.cn

22281046@bjtu.edu.cn

{housaihui, huangyongzhen}@bnu.edu.cn

Abstract

Image Difference Captioning (IDC) has produced many task-specific models, but their gains are often entangled with isolated implementations, data processing, training protocols, and evaluation settings. As a result, the field accumulates method-level scores more easily than reusable structural evidence. This paper proposes **OpenIDC**, a flexible and extensible code-base that reconstructs a unified evidence chain for lightweight task-specific IDC. OpenIDC standardizes data pipelines, model interfaces, training strategies, and evaluation protocols, enabling 8 representative method families with 9 executable variants across 3 benchmarks to be reproduced, compared, decomposed, and diagnosed under consistent conditions. Beyond reproduction, OpenIDC advances IDC from final-score comparison to normalized design-space diagnosis. Based on this diagnosis, we introduce **SIDFormer**, a compact Spatially-guided Invariant-Difference Transformer that serves as a structural answer to lightweight IDC by organizing spatial anchoring, pseudo-change suppression, and reliable difference evidence through functional fusion. Experiments show improved reproducibility and comparability, together with competitive SIDFormer performance across the evaluated benchmarks. Code, reproduced methods, and models will be released.

1 Introduction

Image Difference Captioning (IDC) aims to generate natural language descriptions for before–after image pairs by identifying and explaining semantic changes within a scene. Unlike conventional image captioning, IDC focuses on a change-aware question: what has changed, where it occurs, and how it should be faithfully expressed. This ability is important for scene monitoring, remote sensing, robotic inspection, industrial anomaly observation, and human-centered decision support.

*Corresponding author.

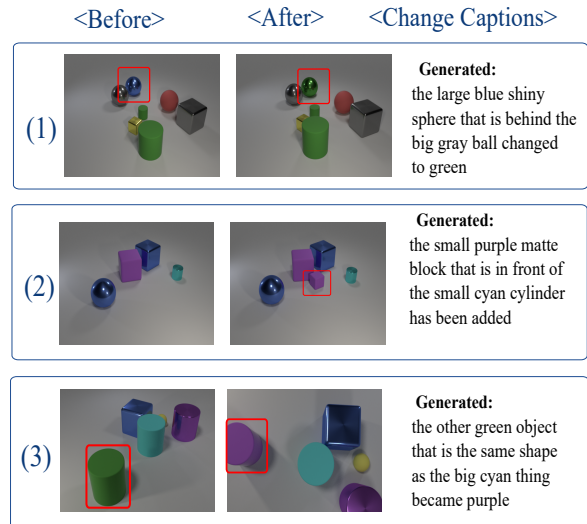


Figure 1: Representative IDC examples. Models must describe semantic changes across before–after image pairs while handling small-object changes, relation changes, and viewpoint-induced distractors.

IDC is a highly coupled multimodal reasoning task: a model must establish cross-image correspondence, locate changed regions or attributes, suppress unchanged content and irrelevant variations such as viewpoint, illumination, scale, and background shifts, and express the true semantic difference in language. Recent studies have advanced IDC from early image-pair comparison and difference-to-caption formulations (Jhamtani and Berg-Kirkpatrick, 2018; Park et al., 2019) to cross-view interaction, fine-grained difference modeling, invariant representation learning, shared-difference disentanglement, and vision-language alignment (Qiu et al., 2021; Tu et al., 2023b,a; Yao et al., 2022). More recently, multimodal large language models and retrieval-augmented reasoning have extended IDC toward open-ended change understanding (Zhang et al., 2024; Jiao et al., 2025; Liu et al., 2025). Nevertheless, lightweight task-specific IDC models remain indispensable: they provide explicit reasoning paths, controllable ob-

jectives, lower reproduction cost, module-level ablation, and traceable failure analysis, making them a necessary testbed for understanding which structures truly improve semantic change captioning.

However, task-specific IDC has reached a critical stage: the field is not short of new model structures, but lacks a unified evidence chain for reliable research conclusions. Here, an *evidence chain* refers to the traceable progression from standardized reproduction and controlled factor comparison to cross-dataset validation and diagnosis-guided model construction. Existing methods are developed in isolated codebases with different preprocessing, feature formats, decoders, objectives, training schedules, and evaluation protocols, making reported gains difficult to interpret as structural evidence. Therefore, this work follows a research path from evidence-chain reconstruction to design-space diagnosis and compact baseline.

To realize this path, we propose **OpenIDC**, a unified and extensible framework for task-specific Image Difference Captioning. OpenIDC standardizes data processing, model invocation, training/testing procedures, loss organization, and evaluation protocols, while preserving method-specific structures as analyzable design factors. Built upon this framework, we further revisit representative IDC branches and derive **SIDFormer**, a compact Spatially-guided Invariant-Difference Transformer, as a structural strong baseline reconstructed from the diagnosed evidence.

This framework-to-model pipeline is organized around the following three aspects.

- **OpenIDC reconstructs the evidence chain of task-specific IDC research.** Instead of treating existing IDC methods as isolated implementations, OpenIDC places representative models under consistent experimental conditions. This allows methods to be reproduced, compared, decomposed, and diagnosed through a unified protocol, so that method-level results can be converted into more trustworthy structural evidence.
- **OpenIDC advances IDC from isolated method competition to normalized design-space diagnosis.** After representative methods are placed in the same framework, OpenIDC further abstracts them into functional design branches according to their roles in change reasoning. These branches include

spatial anchoring, invariant reference, shared-difference disentanglement, interaction and fusion, and decoder-side visual memory. This branch-oriented view allows us to examine which factors are reusable, redundant, or unstable, rather than only comparing complete models as black-box systems.

- **SIDFormer is the structural answer derived from OpenIDC diagnosis.** Based on OpenIDC-based diagnosis, SIDFormer reconstructs lightweight IDC modeling around three core questions: where the change occurs, what should not be regarded as true change, and how reliable difference evidence should be organized for caption generation. It integrates spatial anchoring, invariant-difference representation, and lightweight functional fusion into a concise Transformer-based reasoning path.

Together, OpenIDC and SIDFormer form a pipeline from reproducible comparison to design-space diagnosis and compact baseline construction. The main contributions are summarized as follows:

- **Unified evidence chain.** We propose OpenIDC to standardize data processing, model interfaces, training/testing procedures, and evaluation protocols, enabling representative IDC methods to be reproduced, compared, decomposed, and diagnosed under consistent conditions.
- **Design-space diagnosis.** OpenIDC advances IDC from method-level comparison to factor-level analysis, revealing structural factors such as spatial correspondence, invariant representation, shared-difference disentanglement, robust difference fusion, and cross interaction.
- **Structural strong baseline.** We derive SIDFormer from OpenIDC-based diagnosis, organizing spatial guidance, invariant-difference representation, reliable difference evidence, and functional fusion into a compact Transformer-based reasoning path.

2 Related Work

2.1 Image Difference Captioning

Image Difference Captioning (IDC) was first formulated as an image-pair comparison and difference-to-caption generation task (Jhamtani and Berg-Kirkpatrick, 2018; Park et al., 2019). Later

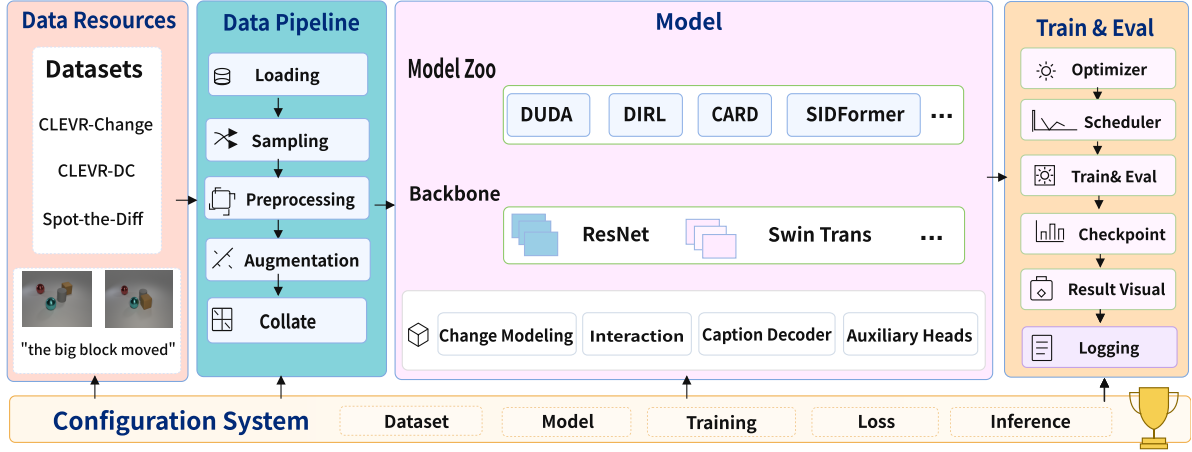


Figure 2: The overall architecture of the OpenIDC codebase.

studies improve this paradigm through attention-based difference localization, cross-view interaction, fine-grained comparison, invariant representation learning, shared-difference disentanglement, and vision-language alignment (Qiu et al., 2021; Tu et al., 2021b, 2023b; Huang et al., 2022; Kim et al., 2021; Tu et al., 2023a; Yao et al., 2022; Guo et al., 2022). Recent works further explore distractor-immune representation learning, context-aware refinement, and MLLM-based open-ended change understanding (Tu et al., 2024a,c; Zhang et al., 2024; Jiao et al., 2025; Liu et al., 2025).

2.2 Reproducibility and Benchmarking

Unified codebases have become important research infrastructure in computer vision, enabling methods to be reproduced, compared, and analyzed under consistent settings (Chen et al., 2019; Fan et al., 2023; Hou et al., 2025). However, IDC still lacks a comparable platform for lightweight task-specific models. To address this gap, OpenIDC standardizes data processing, model interfaces, training/testing procedures, loss management, and evaluation protocols. Beyond reproduction, OpenIDC converts isolated IDC implementations into a normalized design space for controlled comparison, factor-level diagnosis, and strong baseline reconstruction.

3 OpenIDC: A Unified Framework for Image Difference Captioning

The fundamental motivation of OpenIDC is that task-specific Image Difference Captioning has reached a critical stage: the field is not short of new model structures, but lacks a unified evidence chain for reliable research conclusions. Existing IDC methods are usually developed in isolated

codebases, where model design, data processing, feature construction, auxiliary objectives, training schedules, and evaluation protocols are tightly coupled. As a result, reported gains cannot be directly regarded as structural evidence.

OpenIDC is proposed to address this problem by placing representative IDC methods under a unified and transparent protocol. Its goal is not merely to collect existing implementations, but to make IDC methods reproducible, comparable, decomposable, and diagnosable, so that isolated method-level results can be converted into reusable design-space evidence.

3.1 Unified Formulation

As shown in Fig. 2, OpenIDC is designed as a unified and extensible framework for IDC methods. Motivated by the need to reconstruct the evidence chain, OpenIDC standardizes the outer experimental protocol while preserving method-specific structures as controllable components.

Given a before/after image pair (I^b, I^a) and a target caption Y , OpenIDC defines each training sample as

$$\mathcal{E} = \{\mathcal{V}, \mathcal{T}, \mathcal{A}, \mathcal{M}\}, \quad (1)$$

where $\mathcal{V}$ denotes visual inputs, $\mathcal{T}$ denotes textual supervision, $\mathcal{A}$ denotes optional auxiliary annotations, and $\mathcal{M}$ denotes metadata. Here, $\mathcal{V}$ may contain raw images or pre-extracted visual features; $\mathcal{T}$ contains encoded captions, caption lengths, masks, and ignored labels; $\mathcal{A}$ contains method-specific supervision such as negative samples, part-of-speech labels, keyword/tag labels, or dependency-parsing labels; and $\mathcal{M}$ records sample identifiers and auxiliary information. This definition allows heterogeneous IDC methods to be expressed through the

same sample structure without discarding their original modeling requirements.

A model instantiated in OpenIDC follows a unified forward interface:

$$\mathbf{O}, \mathbf{Z} = f_{\theta}(\mathcal{E}), \quad (2)$$

where $\mathbf{O}$ denotes captioning outputs and $\mathbf{Z}$ denotes intermediate representations used for auxiliary objectives, visualization, or structural diagnosis. Different methods may retain their own change detector, attention module, relation module, disentangle module, decoder, or auxiliary head, but they are invoked through the same outer interface.

OpenIDC further unifies loss organization and execution protocol. Language modeling loss provides the basic caption supervision, while attention regularization, tag loss, consistency loss, alignment loss, and other method-specific objectives are explicitly declared and aggregated through configuration files. At the execution level, OpenIDC standardizes data loading, logging, checkpoint management, scheduler interface, inference procedure, and caption evaluation scripts.

3.2 Revisiting Design Factors under OpenIDC

Based on the unified architecture above, OpenIDC revisits representative IDC methods by decomposing them along the evidence chain of change captioning. Although existing methods are proposed as complete systems, their core designs usually intervene in one of several recurring reasoning steps: establishing before/after correspondence, filtering pseudo changes, separating shared content from difference evidence, exchanging and fusing cross-view information, and conditioning the language decoder on visual evidence. We therefore extract characteristic design branches from these recurring roles rather than treating each method as an indivisible model.

Specifically, designs for establishing correspondence are summarized as spatial and positional branches; designs for suppressing viewpoint, illumination, or background distractions are summarized as invariant representation branches; designs for separating unchanged semantics from describable changes are summarized as shared-difference branches; designs for cross-view reasoning and evidence integration are summarized as interaction and fusion branches; and designs controlling what visual memory is exposed to the caption decoder are summarized as decoder memory branches.

Spatial and positional branches mainly ad-

dress the question of where the change occurs. Methods such as DUDA (Park et al., 2019), MCCFormers (Qiu et al., 2021), and NCT (Tu et al., 2023b) indicate that IDC often requires position-sensitive correspondence between before/after images, especially for object movement, spatial relation changes, and small-object attribute changes. Change localization, coordinate augmentation, token positional encoding, and local neighborhood modeling all provide spatial cues. However, these cues are not equally necessary in a compact baseline: explicit coordinates provide a lightweight spatial prior, while additional positional tokens or heavy local branches may introduce redundant spatial modeling.

Invariant representation branches address the question of what should not be regarded as true change. Methods such as VARD (Tu et al., 2023a) and VACC (Kim et al., 2021) show that direct before/after comparison may incorrectly amplify viewpoint, illumination, scale, occlusion, or background variations. A stable invariant reference helps identify semantics shared by the two views and prevents pseudo-change responses from entering the final difference evidence. Thus, invariant modeling is retained not to suppress true changes, but to filter irrelevant visual disturbances.

Shared-difference branches address the question of which difference evidence is worth describing. Methods such as NCT (Tu et al., 2023b), VARD (Tu et al., 2023a) suggest that before/after images usually share most visual content, while only a small part corresponds to describable changes. Separating shared semantics from change-sensitive residuals helps avoid exposing mixed or redundant features to the decoder. This moves IDC from raw feature differencing toward reliable change-evidence organization.

Interaction and fusion branches determine how before/after evidence is exchanged and integrated. Methods such as MCCFormers (Qiu et al., 2021), R³Net (Tu et al., 2021a) introduce cross-view interaction, local neighborhood comparison, relation reasoning, or multi-source fusion to enhance fine-grained change understanding. However, these branches may overlap with spatial anchoring or shared-difference modeling, and excessive interaction paths may increase optimization difficulty. Therefore, interaction should mainly support compact evidence fusion rather than expand the model with multiple parallel branches.

Decoder memory branches determine what

visual information is exposed to the language generator. Methods such as DUDA (Park et al., 2019), SMART (Tu et al., 2024b) explore different decoder structures or context refinement mechanisms. While additional visual memories may enrich generation, excessive memory sources can dilute change evidence and make the decoder attend to unchanged or irrelevant regions. Therefore, IDC decoding should rely on organized difference evidence rather than all intermediate visual representations.

Overall, the above analysis shows that existing IDC methods have explored a series of useful design branches, but these branches are not simply complementary to each other. Spatial cues, invariant representations, shared-difference modeling, interaction modules, and decoder memories may address different aspects of change understanding, yet they can also introduce overlapping functions, redundant information paths, and unstable evidence organization when directly stacked together. Therefore, the key issue is no longer whether more branches can be added, but how to reorganize the effective factors into a more compact and reliable reasoning structure. This observation motivates the model reconstruction in Section 4.

4 SIDFormer

Based on the OpenIDC-based diagnosis in Sec. 3, we propose SIDFormer, a compact Spatially-guided Invariant-Difference Transformer for lightweight task-specific IDC. SIDFormer follows the structural principle derived from the design-space analysis: reliable IDC modeling should first establish spatially sensitive correspondence, then suppress pseudo-change responses through invariant semantic reference, and finally organize the remaining difference evidence into a compact decoder-friendly representation.

4.1 Architecture

Based on the OpenIDC-based design diagnosis, SIDFormer is constructed as a compact change-evidence reasoning path rather than a collection of independent branches. Its architecture follows four functional stages: spatially-guided visual tokenization, invariant semantic reference construction, shared-difference disentanglement, and reliable difference evidence fusion. The overall architecture is shown in Fig. 3. These stages correspond to the three core requirements of IDC: locat-

ing where the change occurs, suppressing pseudo changes, and organizing describable difference evidence for caption generation.

Spatially-guided visual tokenization. Given a before/after image pair, a shared visual feature extractor first produces feature maps $\mathbf{X}^b, \mathbf{X}^a \in \mathbb{R}^{B \times C \times H \times W}$, where b and a denote the before and after views. Since IDC often involves object movement, spatial relation changes, and small-object attribute changes, purely semantic visual features are insufficient for reliable cross-view comparison. SIDFormer therefore injects normalized spatial coordinates into both views:

$$\begin{aligned} \tilde{\mathbf{X}}^b &= [\mathbf{X}^b; \mathbf{P}], \quad \tilde{\mathbf{X}}^a = [\mathbf{X}^a; \mathbf{P}], \\ \mathbf{P}_{i,j} &= \left(\frac{2j}{W} - 1, \frac{2i}{H} - 1 \right). \end{aligned} \quad (3)$$

The coordinate-enhanced features are then flattened and projected into visual tokens $\mathbf{V}^b, \mathbf{V}^a \in \mathbb{R}^{B \times N \times d}$, where $N = H * W$. This design provides a lightweight spatial anchor for before/after correspondence without introducing an additional heavy localization branch.

Invariant semantic reference. Direct feature differencing may amplify viewpoint, illumination, scale, or background variations that should not be described as semantic changes. To reduce such pseudo-change responses, SIDFormer constructs an invariant semantic reference from the two views:

$$\mathbf{R} = \text{Gate} \left(\mathbf{V}^b, \mathbf{V}^a, \phi_s((\mathbf{V}^b + \mathbf{V}^a)/2) \right), \quad (4)$$

where $\phi_s(\cdot)$ maps the averaged before/after representation into a shared semantic space, and $\text{Gate}(\cdot)$ denotes a learnable sigmoid-gated fusion module. Specifically, the gate predicts element-wise sigmoid weights from the view-specific tokens and their shared semantic center, and uses these weights to balance persistent invariant content against view-dependent responses. Let $\mathbf{I}^b$ and $\mathbf{I}^a$ denote the corresponding normalized token representations produced by the invariant branch before gated fusion. The resulting invariant reference $\mathbf{R}$ serves as a stable semantic anchor, helping the model distinguish persistent scene content from unreliable visual fluctuations. In this way, SIDFormer does not simply maximize visual differences, but first establishes what should remain stable across the two images.

Shared-difference disentanglement. Although before/after images differ in specific regions, most scene content remains unchanged. If shared semantics and true changes are mixed and directly exposed to the decoder, the generated caption may

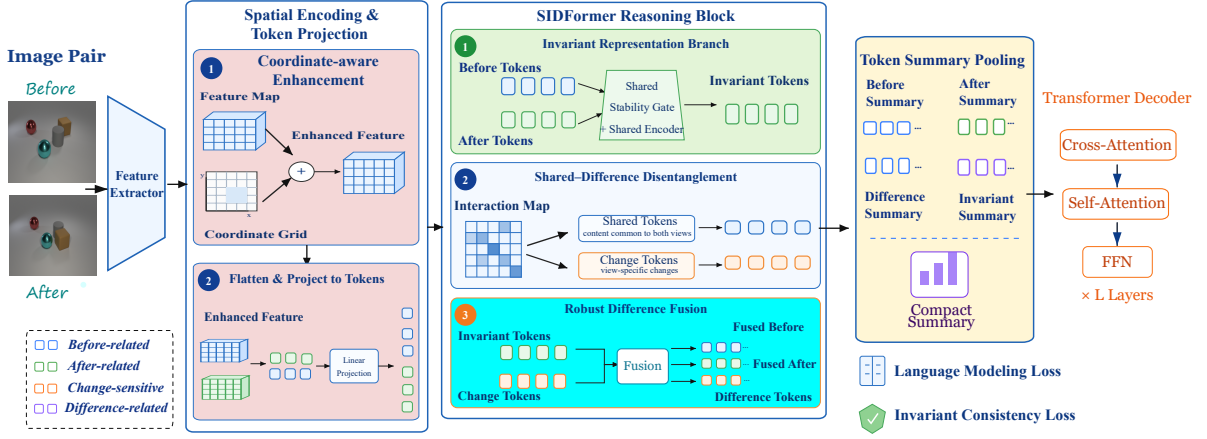


Figure 3: The overall architecture of the proposed SIDFormer model.

attend to unchanged regions or describe redundant information. SIDFormer therefore estimates cross-view shared components and uses their residuals as change-sensitive evidence:

$$\begin{aligned} \mathbf{U}^v &= \text{Attn}(\mathbf{V}^v, \mathbf{V}^{\bar{v}}), \\ \Delta^v &= \mathbf{V}^v - \mathbf{U}^v, \quad v \in \{b, a\}, \end{aligned} \quad (5)$$

where $\bar{v}$ denotes the opposite view of v . Here, $\mathbf{U}^v$ represents the cross-view shared component retrieved from the opposite view, while Δ^v captures the residual evidence that cannot be explained by shared semantics. This step converts raw before/after comparison into a more structured shared-difference representation.

Reliable difference evidence fusion. The invariant reference and residual evidence are further fused to produce robust before/after representations $\mathbf{F}^b$ and $\mathbf{F}^a$. Rather than relying on a single difference form, SIDFormer combines global and token-level difference cues:

$$\begin{aligned} \mathbf{D} &= \phi_{\text{diff}}([\mathbf{D}_g; \mathbf{D}_l]) + \mathbf{D}_0, \\ \mathbf{D}_g &= \text{Pool}(\mathbf{F}^a) - \text{Pool}(\mathbf{F}^b), \\ \mathbf{D}_l &= \mathbf{F}^a - \mathbf{F}^b. \end{aligned} \quad (6)$$

Here, $\mathbf{D}_g$ captures image-level change tendency and is broadcast across the N token positions before concatenation, $\mathbf{D}_l$ preserves token-level local difference evidence, and $\mathbf{D}_0$ denotes an invariant-guided difference seed. The difference seed $\mathbf{D}_0$ is obtained by projecting the fused representation of gated invariant tokens and change-sensitive residual tokens. It provides an initial change-aware reference that complements the global and token-level difference cues, rather than introducing an additional reasoning branch. The projection $\phi_{\text{diff}}(\cdot)$ integrates these cues into decoder-friendly difference tokens. This fusion stage allows SIDFormer

to preserve both global change context and local change details while avoiding excessive parallel branches.

Caption generation. The final difference-aware representation $\mathbf{D}$ is used as a compact visual memory for a Transformer decoder. SIDFormer encourages the decoder to focus on semantic differences that are both visually grounded and linguistically describable, thereby connecting the visual reasoning path with the final change caption generation.

4.2 Objective

SIDFormer is optimized with caption generation loss and lightweight invariant consistency regularization. The two terms serve complementary purposes: the former supervises change-caption generation, while the latter stabilizes the shared semantic representation across the before and after views:

$$\mathcal{L} = \mathcal{L}_{\text{cap}} + \lambda_{\text{con}} \mathcal{L}_{\text{con}}. \quad (7)$$

Here, $\mathcal{L}_{\text{cap}}$ denotes the standard caption generation loss:

$$\mathcal{L}_{\text{cap}} = -\frac{1}{|\Omega|} \sum_{t \in \Omega} \log p_{\theta}(y_t | y_{<t}, \mathbf{D}), \quad (8)$$

where Ω denotes valid target positions and $\mathbf{D}$ denotes the difference-aware decoder memory. The consistency term is implemented as a token-wise cosine distance between corresponding before/after invariant representations:

$$\mathcal{L}_{\text{con}} = \frac{1}{BN} \sum_{m=1}^B \sum_{i=1}^N \left(1 - (\mathbf{I}_{m,i}^b)^\top \mathbf{I}_{m,i}^a\right). \quad (9)$$

where B and N denote the batch size and number of visual tokens, respectively, while m and i index samples and visual tokens. We set $\lambda_{\text{con}} = 0.05$ in all experiments.

Table 1: Comparison on three IDC benchmarks. “Orig.” denotes original reported results, and “OpenIDC” denotes reproduced or additionally evaluated results under our unified framework. The reproduced collection contains eight method families and nine executable variants because SMART is instantiated with both LSTM and Transformer decoders. A dash indicates unavailable results.

Method	Source	CLEVR-Change					Spot-the-Diff					CLEVR-DC				
		B-4	M	R-L	C	S	B-4	M	R-L	C	S	B-4	M	R-L	C	S
DUDA	Orig.	47.3	33.9	–	112.3	24.5	8.1	11.8	29.1	32.5	–	40.3	27.1	–	56.7	16.1
	OpenIDC	53.6	37.5	70.7	116.5	31.5	8.9	12.3	32.0	35.5	17.4	46.1	31.5	64.0	69.8	14.8
SRDRL	Orig.	54.9	40.2	73.3	122.2	32.9	–	13.0	31.0	35.3	18.0	–	–	–	–	–
	OpenIDC	56.0	40.3	73.2	123.8	33.0	8.4	11.9	28.6	34.0	17.2	46.5	31.7	64.2	72.8	15.2
R ³ Net	Orig.	54.7	39.8	73.1	123.0	32.6	–	13.1	32.6	36.6	18.8	–	–	–	–	–
	OpenIDC	53.5	39.6	72.2	121.0	32.3	8.1	13.0	31.8	36.6	17.7	51.6	30.4	64.5	78.9	16.2
NCT	Orig.	55.1	40.2	73.8	124.1	32.9	–	–	–	–	–	47.5	32.5	65.1	76.9	15.6
	OpenIDC	54.9	41.0	73.3	123.8	33.5	8.0	11.7	30.0	30.4	17.6	47.5	32.6	64.9	76.9	15.3
VARD-LSTM	Orig.	55.2	40.8	74.1	124.1	33.3	–	13.1	33.1	39.3	17.5	48.4	32.0	–	72.4	14.2
	OpenIDC	55.6	40.8	73.5	124.0	33.6	10.3	12.6	32.0	39.0	16.9	52.6	32.8	66.8	84.3	16.8
SMART-LSTM	Orig.	55.7	40.6	73.8	126.5	33.4	13.1	–	32.8	40.0	19.5	–	–	–	–	–
	OpenIDC	56.1	41.1	74.4	125.3	33.2	11.0	11.8	30.5	39.6	18.9	53.8	32.0	66.0	86.3	16.6
SMART-Trans.	Orig.	56.1	40.8	74.2	127.0	33.4	13.5	–	31.6	39.4	19.0	–	–	–	–	–
	OpenIDC	55.9	41.2	74.5	126.5	33.5	12.0	11.3	30.4	37.9	18.6	51.8	32.5	66.6	86.1	16.9
CARD	Orig.	–	–	–	–	–	6.6	10.8	26.9	32.4	17.8	–	–	–	–	–
	OpenIDC	55.8	40.9	74.4	124.4	32.8	9.6	12.5	31.5	40.2	16.8	52.5	32.6	67.0	85.9	16.2
DIRL	Orig.	54.6	38.1	71.9	123.6	31.8	10.3	13.8	32.8	40.9	19.9	51.4	32.3	66.3	84.1	16.8
	OpenIDC	56.0	41.0	74.4	125.6	33.0	11.1	13.2	33.3	39.6	18.2	54.4	32.8	67.1	84.6	17.6
SIDFormer	OpenIDC	59.4	42.0	76.1	137.5	34.0	12.2	14.1	34.9	42.3	22.1	55.2	34.1	68.0	86.7	18.4

5 Experiments

5.1 Experimental Setup

We conduct experiments on three benchmarks integrated in OpenIDC: **CLEVR-Change** (Park et al., 2019), **CLEVR-DC** (Kim et al., 2021), and **Spot-the-Diff** (Jhamtani and Berg-Kirkpatrick, 2018). CLEVR-Change contains 79,606 synthetic image pairs with 493,735 captions, covering color, texture, add, drop, move, and viewpoint distractors. CLEVR-DC contains 48,000 image pairs and focuses on stronger viewpoint variations, making it suitable for evaluating pseudo-change robustness. Spot-the-Diff contains 13,192 real-world surveillance image pairs, providing a more realistic setting with irregular scene changes and background noise. Together, these benchmarks cover controlled synthetic changes, viewpoint-induced pseudo changes, and real-world visual differences.

The compared task-specific IDC methods are listed in Table 1. The reproduced collection contains eight method families and nine executable

variants; SMART-LSTM and SMART-Trans. share the same method family but use different caption decoders. Following the unified OpenIDC protocol, all reproduced methods share the same data interface, vocabulary construction, inference procedure, logging mechanism, checkpoint management, and evaluation scripts. Method-specific modules, auxiliary objectives, optimizer choices, and decoder variants are explicitly controlled by configuration files, so that architectural differences remain visible while the surrounding experimental protocol is transparent and reproducible.

We report BLEU-4, METEOR, ROUGE-L, CIDEr, and SPICE (Papineni et al., 2002; Banerjee and Lavie, 2005; Lin, 2004; Vedantam et al., 2015; Anderson et al., 2016) using the Microsoft COCO caption evaluation toolkit (Chen et al., 2015). Unless otherwise specified, reproduced methods are trained with an iteration-based StepLR schedule, using a step size of 4000, a decay factor of 0.5, and a 1000-iteration warm-up from 1×10^{-5} . For SID-

Former, we use AdamW with an initial learning rate of 2×10^{-4} and a weight decay of 1×10^{-4} and set the consistency weight λ_{con} to 0.05. All results are obtained from the OpenIDC inference and evaluation pipeline without modifying metric scripts for individual methods.

This unified setup ensures that the reported differences mainly reflect model design and explicitly configured objectives, rather than hidden changes in data processing or metric computation.

5.2 Main Results

OpenIDC reproduction under a unified protocol.

We first evaluate whether OpenIDC can reproduce and compare representative task-specific IDC methods under a unified protocol. As shown in Table 1, OpenIDC reports both originally published results when available and reproduced results obtained with the same data interface, inference procedure, and evaluation scripts. Since original implementations may differ in preprocessing, feature extraction, training schedules, decoding strategies, and hidden engineering details, our goal is not exact numerical equivalence, but a controlled space for re-observing existing methods. More specifically, the observed gaps can arise from differences in dataset filtering and caption organization, visual feature checkpoints and normalization, vocabulary construction and auxiliary supervision, optimization schedules and checkpoint selection, as well as decoding and metric implementations. The original scores are therefore retained as historical reference points, whereas the OpenIDC scores provide a common and traceable comparison protocol; OpenIDC is not intended to systematically increase the original scores or enforce numerical identity with every published implementation.

SIDFormer as a compact structural baseline. Table 1 also shows that SIDFormer achieves competitive performance across CLEVR-Change, CLEVR-DC, and Spot-the-Diff. Although it does not integrate very recent non-open-source components or MLLM-based open-ended techniques, SIDFormer remains competitive with the strongest reproduced task-specific methods and recent methods reported in the literature. This suggests that reliable evidence organization can provide a compact and reproducible strong baseline without heavy branch expansion.

Table 2: Ablation analysis of SIDFormer on CLEVR-Change.

Variant	B-4	M	R-L	C	S
<i>Construction path</i>					
Baseline	51.7	40.3	72.9	127.4	31.8
+ Spatial	52.1	39.4	73.0	128.8	32.4
+ Invariant ref.	53.5	40.5	74.2	129.7	32.0
+ Shared-diff.	55.6	40.9	75.0	133.0	33.2
+ Diff. fusion	58.1	41.8	76.0	136.6	33.9
Full SIDFormer	59.4	42.0	76.1	137.5	34.0
<i>Removal and substitution</i>					
w/o Cons. reg.	58.1	41.8	76.0	133.0	33.2
w/o Invariant ref.	58.0	41.8	75.9	135.8	33.4
w/o Shared-diff.	56.7	41.0	74.8	134.7	32.7
w/o Diff. fusion	56.5	41.1	74.6	135.5	33.5
w/ LSTM decoder	54.2	35.9	65.1	117.5	28.5

5.3 Ablation Study

We conduct ablation studies from two complementary perspectives. First, we analyze the construction path of SIDFormer and remove each retained component to verify whether the final evidence chain is effective. Second, we examine several candidate branches that are plausible in the IDC design space but are not retained in the final architecture. Together, these experiments explain not only what SIDFormer keeps, but also why it avoids unnecessary branch expansion.

Retained component analysis. The baseline uses the same data pipeline, feature extractor, vocabulary, training protocol, and evaluation metrics as SIDFormer, but removes spatial coordinate augmentation, invariant reference, shared-difference disentanglement, robust fusion, and consistency regularization. As shown in Table 2, each retained component contributes to the final model. Spatial encoding improves position-sensitive correspondence, invariant reference helps suppress pseudo changes, shared-difference disentanglement separates common semantics from change evidence, and robust fusion organizes difference tokens for caption generation. Removing these components consistently weakens performance, while replacing the Transformer decoder with an LSTM decoder causes a clear drop.

Candidate branch analysis. We further evaluate candidate branches including token positional encoding, local branch, dual cross attention, context memory, invariant memory, and max-pooling global difference. These branches correspond to reasonable extensions in the IDC design space,

Table 3: Candidate branch analysis on CLEVR-Change. All candidate variants are tested but not retained in SIDFormer.

Variant	B-4	M	R-L	C	S
Full SIDFormer	59.4	42.0	76.1	137.5	34.0
+ Token pos. enc.	58.9	42.1	76.0	137.0	33.7
+ Local branch	30.7	29.6	49.2	69.5	17.0
+ Dual cross attn.	57.9	41.6	75.1	135.8	33.5
+ Local + dual cross	46.1	37.4	68.5	116.9	28.1
+ Context memory	58.2	41.5	75.7	135.0	32.5
+ Invariant memory	56.9	41.3	74.9	134.4	33.3
+ Max-pool diff.	58.0	41.5	75.4	135.1	32.9

but they do not consistently improve SIDFormer. As shown in Table 3, some variants bring only marginal single-metric gains, while others lead to clear performance degradation. This indicates that increasing branch complexity does not necessarily produce more reliable change evidence. Therefore, SIDFormer retains the compact evidence organization path and treats these branches as design hypotheses rather than default components. Additional validation of the same candidate branches on CLEVR-DC and Spot-the-Diff is reported in Appendix A.

6 Conclusion

This paper presents **OpenIDC**, a unified infrastructure that reconstructs the evidence chain of task-specific IDC. Built on OpenIDC, we revisit representative methods under consistent conditions and move IDC from final-score comparison to design-space diagnosis. Based on the resulting structural evidence, we introduce **SIDFormer**, a compact Transformer-based baseline that organizes spatial correspondence, invariant context, and reliable difference evidence for faithful caption generation. Experiments show that OpenIDC provides a reproducible foundation for cumulative IDC research, while SIDFormer achieves competitive performance without heavy branch expansion.

Limitations

OpenIDC currently focuses on lightweight task-specific IDC models. Recent MLLM-based and open-ended change understanding methods involve different prompting, retrieval, training, and evaluation protocols, and are therefore beyond the current scope. The diagnostic evidence is also bounded by the evaluated benchmarks. CLEVR-Change and

CLEVR-DC contain controlled synthetic scenes, while Spot-the-Diff uses real images but still represents a curated set of relatively localized changes. Consequently, observations concerning positional cues, interaction branches, decoder memories, or global-difference pooling may depend on the visual complexity and change distributions of these datasets. The cross-dataset candidate-branch analysis reduces reliance on a single benchmark, but it does not establish universal conclusions for unconstrained surveillance, medical imaging, remote sensing, or other open-world settings. The diagnosed factors should therefore be interpreted as evidence under the current unified protocol rather than as dataset-independent architectural rules. Nevertheless, OpenIDC first establishes a reproducible and diagnosable foundation for conventional task-specific IDC, which can serve as a starting point for extending unified analysis to these emerging paradigms.

Ethical Considerations

This work focuses on reproducible benchmarking and lightweight task-specific image difference captioning. We do not redistribute the original datasets. Users should obtain CLEVR-Change, CLEVR-DC, and Spot-the-Diff from their official sources and follow the corresponding licenses or terms of use. The OpenIDC code and models, once released, are intended for research use. The experiments are conducted on publicly available benchmarks. We do not introduce new human-subject data or personally identifiable information. Nevertheless, image difference captioning systems may be applied in monitoring or surveillance scenarios, and their predictions should not be used as sole evidence in safety-critical decision making without human verification.

Acknowledgments

This work was jointly supported by the Joint Fund for the Provincial Science and Technology R&D Program of Henan Province under Grant No. 245200810009, the National Natural Science Foundation of China under Grant Nos. 62476027 and 62276025, and the Fundamental Research Funds for the Central Universities under Grant No. 2253200026.

References

- Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. SPICE: Semantic propositional image caption evaluation. In *Proceedings of the European Conference on Computer Vision*, pages 382–398.
- 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. Association for Computational Linguistics.
- Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, and 6 others. 2019. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*.
- Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. 2015. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.
- Chao Fan, Junhao Liang, Chuanfu Shen, Saihui Hou, Yongzhen Huang, and Shiqi Yu. 2023. Opengait: Revisiting gait recognition towards better practicality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9707–9716.
- Zixin Guo, Tzu-Jui Julius 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. Association for Computational Linguistics.
- Saihui Hou, Panjian Huang, Zengbin Wang, Yuan Liu, Zeyu Li, Man Zhang, and Yongzhen Huang. 2025. Openanimals: Revisiting person re-identification for animals towards better generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14369–14379.
- Qingbao Huang, Yu Liang, Jielong Wei, Yi Cai, Hanyu Liang, Ho-fung Leung, and Qing Li. 2022. Image difference captioning with instance-level fine-grained feature representation. *IEEE Transactions on Multimedia*, 24:2004–2017.
- 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.
- Qirui Jiao, Daoyuan Chen, Yilun Huang, Bolin Ding, Yaliang Li, and Ying Shen. 2025. Img-Diff: Contrastive data synthesis for multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9296–9307.
- Hoeseong Kim, Jongseok Kim, Hyungseok Lee, Hyun-sung Park, and Gunhee Kim. 2021. Viewpoint-agnostic change captioning with cycle consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2095–2104.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.
- Yuan Liu, Saihui Hou, Saijie Hou, Jiabao Du, Shibe Meng, and Yongzhen Huang. 2025. OmniDiff: A comprehensive benchmark for fine-grained image difference captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21440–21449.
- 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. Association for Computational Linguistics.
- 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.
- 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.
- Yunbin Tu, Liang Li, Li Su, Junping Du, Ke Lu, and Qingming Huang. 2023a. Viewpoint-adaptive representation disentanglement network for change captioning. *IEEE Transactions on Image Processing*, 32:2620–2635.
- Yunbin Tu, Liang Li, Li Su, Ke Lu, and Qingming Huang. 2023b. Neighborhood contrastive transformer for change captioning. *IEEE Transactions on Multimedia*, 25:9518–9529.
- 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.
- 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*, 46(7):4926–4943.

Yunbin Tu, Liang Li, Li Su, Zheng-Jun Zha, Chenggang Yan, and Qingming Huang. 2024c. Context-aware difference distilling for multi-change captioning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 7941–7956. Association for Computational Linguistics.

Yunbin Tu, Liang Li, Chenggang Yan, Shengxiang Gao, and Zhengtao Yu. 2021a. R^3 Net: 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.

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.

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.

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.

A Why Additional Branches Are Not Retained

This appendix provides additional explanation for the candidate branch analyses reported in the main text. These variants are tested during the reconstruction of SIDFormer to examine whether increasing architectural complexity can further improve IDC performance. The goal is not to exhaustively enumerate all possible combinations, but to verify whether the final SIDFormer design retains stable and transferable structural factors rather than simply stacking more branches.

In the final configuration, SIDFormer keeps spatial coordinate augmentation, invariant semantic reference, shared-difference disentanglement, mean-pooled global difference modeling, compact decoder memory, and Transformer-based caption decoding. Several additional branches are considered as design hypotheses, including token positional encoding, local interaction branches, dual

cross attention, decoder context memory, decoder invariant memory, and alternative global difference pooling. As reported in the main text, these variants either bring only marginal single-metric gains or cause unstable performance degradation while increasing architectural complexity. Therefore, they are not retained in the final architecture. Table 5 further evaluates the same candidate branches on CLEVR-DC and Spot-the-Diff. None of the additional branches consistently improves the full model across the two datasets. Some variants remain competitive on individual metrics, but their effects are not stable across datasets, supporting the compact configuration while also indicating that the diagnosis should be interpreted within the evaluated benchmark scope.

Token positional encoding. Token positional encoding is tested because spatial correspondence is important for IDC. However, SIDFormer already injects explicit coordinate information into before/after feature maps. Adding token-level positional encoding introduces another spatial prior, but does not consistently improve the final results. This suggests that explicit coordinate augmentation is sufficient for providing lightweight spatial anchoring in the current compact design.

Local interaction and dual cross attention. Local interaction branches and dual cross attention are evaluated to test whether stronger cross-view interaction can improve fine-grained difference modeling. Although these branches are intuitively useful, they also introduce additional parameters, interaction paths, and optimization difficulty. The results show that heavier local or dual-cross structures do not provide stable gains under the unified protocol, and in some cases substantially degrade performance. This indicates that more complex interaction does not necessarily lead to more reliable difference evidence.

Decoder memory variants. Decoder context memory and invariant memory are tested to examine whether exposing more visual sources to the language decoder can improve caption generation. However, IDC requires the decoder to focus on organized change evidence rather than all intermediate visual representations. Additional memory sources may dilute the difference evidence and make the decoder attend to unchanged or weakly relevant regions. Therefore, SIDFormer uses compact difference memory as the final decoder input.

Table 4: Supported methods and integration status in OpenIDC.

Method / Variant	Implementation basis	OpenIDC status	Released-code datasets	Main design factor
DUDA	Official / Adapted	Reproduced	C-Change	Dual attention difference modeling
SRDRL	Official / Adapted	Reproduced	C-Change	Semantic relation and difference reasoning
R ³ Net	Official / Adapted	Reproduced	C-Change	Relation refinement
NCT	Official / Adapted	Reproduced	C-Change / C-DC	Neighborhood-aware interaction
VARD-LSTM	Official / Adapted	Reproduced	C-Change	Viewpoint-adaptive representation
SMART-LSTM	Official / Adapted	Reproduced	C-Change	Shared-memory reasoning with LSTM decoder
SMART-Trans.	Official / Adapted	Reproduced	C-Change	Shared-memory reasoning with Transformer decoder
CARD	Official / Adapted	Reproduced	Spot	Context-aware refinement
DURL	Official / Adapted	Reproduced	C-Change / C-DC	Distractor-immune representation learning
SIDFormer	Ours	Proposed	C-Change / Spot / C-DC	Spatially-guided invariant-difference reasoning

Table 5: Cross-dataset validation of candidate branches on CLEVR-DC (DC) and Spot-the-Diff (StD). C and S denote CIDEr and SPICE, respectively.

Variant	DC C	DC S	StD C	StD S
Token positional	81.08	15.51	31.21	16.17
Local interaction	84.42	16.81	36.00	18.90
Dual cross attention	85.91	17.16	39.80	20.80
Local + dual	83.96	16.29	34.50	18.10
Context memory	85.53	18.03	39.20	20.20
Invariant memory	84.21	16.06	38.70	20.40
Max-pool difference	86.12	18.23	39.50	20.50
Full SIDFormer	86.65	18.41	42.30	22.06

Global difference pooling. We also test alternative global difference pooling. Compared with max pooling, mean-pooled global difference provides a more stable summary of scene-level change tendency. Max pooling may over-emphasize sparse high-response regions, which can be harmful when local activations are caused by noise or pseudo changes. Thus, mean pooling is retained in the final model.

Overall, the candidate analysis supports the central design principle of SIDFormer: the final model should preserve stable functional roles, including spatial anchoring, invariant reference, shared-difference evidence, and robust fusion, while avoiding unnecessary branch expansion. This makes SIDFormer a compact structural baseline rather than an accumulated multi-branch architecture.

B How OpenIDC Reproduces Existing IDC Methods

This section summarizes the reproduction protocol and supported methods in OpenIDC. The purpose of OpenIDC is not to simply collect reported numbers from existing papers, but to place representative IDC methods into a unified and controllable

experimental environment. For each reproduced method, OpenIDC preserves its method-specific architecture as much as possible, while standardizing the surrounding data loading, sample construction, training loop, checkpoint management, inference procedure, and caption evaluation protocol.

B.1 Result Types in OpenIDC

In the main comparison table, we report two types of results. *Orig.* denotes the scores reported in the original papers. These results are used as reference points, but they may be obtained under different preprocessing pipelines, feature formats, training schedules, decoding strategies, or evaluation implementations. *OpenIDC* denotes results reproduced or additionally evaluated within our unified framework. These results are produced under the OpenIDC data interface, model invocation format, training/evaluation loop, and metric computation pipeline.

B.2 Supported Methods

OpenIDC currently supports representative task-specific IDC methods that are suitable for unified reproduction and structural diagnosis. These methods are selected according to four criteria: public availability of sufficient methodological details, feasibility of implementation under a shared framework, architectural diversity, and relevance to task-specific IDC. Together, they cover major design routes in IDC, including attention-based difference localization, relation reasoning, neighborhood interaction, viewpoint-adaptive representation learning, context-aware refinement, distractor-immune learning, and Transformer-based caption decoding.

Table 4 summarizes the supported methods and their integration status. Some methods are represented by more than one model variant, such as SMART-LSTM and SMART-Trans., because they use different caption decoders under related visual

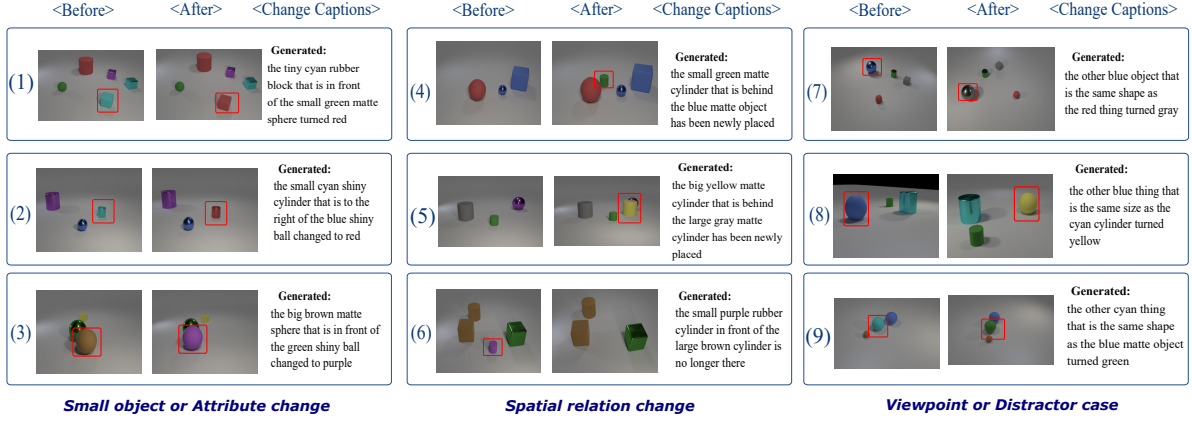


Figure 4: Additional qualitative examples where SIDFormer correctly describes semantic changes. These examples cover small-object changes, attribute changes, spatial relation changes, and distractor cases.

reasoning designs. Therefore, OpenIDC covers 8 representative method families with 9 reproduced model variants in the current version.

B.3 Unified Reproduction Protocol

For OpenIDC reproduced results, all methods are evaluated under a unified caption evaluation protocol. Specifically, datasets are accessed through the same data interface, captions are tokenized using the corresponding vocabulary files, and predictions are evaluated with the same captioning metrics. For methods that require auxiliary labels such as part-of-speech tags, ignored labels, negative samples, or tag-level supervision, OpenIDC keeps these method-specific signals in the configuration while maintaining the same outer training and evaluation pipeline.

The unified protocol contains three levels. First, the **data level** standardizes dataset splits, caption loading, sequence construction, and batch sampling. Second, the **model level** provides a common invocation interface so that different IDC architectures can be trained and evaluated through the same framework. Third, the **evaluation level** standardizes checkpoint loading, decoding, and metric computation. This design makes the comparison more transparent because differences between methods are mainly attributed to their architectural designs and explicitly configured objectives, rather than hidden implementation details.

B.4 Treatment of Method-specific Components

Although OpenIDC standardizes the outer training and evaluation pipeline, it does not force all methods to share the same internal architecture

Table 6: Examples of method-specific components preserved in OpenIDC.

Component type	Examples
Visual reasoning	Attention, relation reasoning, local interaction
Representation learning	Invariant learning, disentanglement, distractor suppression
Decoder type	LSTM decoder, Transformer decoder
Auxiliary supervision	Tag loss, attention regularization, consistency loss
Input signals	Negative samples, POS labels, ignore labels
Evaluation protocol	Unified metric computation and checkpoint loading

or the same loss composition. Instead, method-specific components are explicitly declared in configuration files. For example, a method may use an LSTM decoder or a Transformer decoder; it may use attention regularization, tag prediction, consistency constraints, or other auxiliary objectives when required by the original design. OpenIDC keeps these components visible and configurable, so that reproduced results can be traced back to explicit design choices.

This treatment is important for IDC because many reported gains in prior methods may come from a mixture of architectural design, auxiliary supervision, preprocessing, decoder choice, and training protocol. By making these factors explicit, OpenIDC supports not only reproduction but also design-space diagnosis. In other words, OpenIDC does not merely answer whether one complete method is stronger than another; it also helps identify which structural factors are stable and reusable across methods.

C Qualitative Analysis

This section provides additional qualitative examples to further illustrate the behavior of SIDFormer under the OpenIDC evaluation protocol. The main paper reports representative examples,

while this appendix includes more cases covering small-object changes, attribute changes, spatial relation changes, viewpoint-induced distractors, and real-world surveillance differences. These examples help explain why organizing reliable difference evidence is important for faithful image difference captioning.

Fig. 4 shows additional successful examples. Compared with the baseline and representative reproduced methods, SIDFormer more consistently focuses on semantically meaningful changes rather than unchanged background regions or viewpoint-induced distractors. For small-object changes, the spatially enhanced representation helps the model maintain location-sensitive before/after correspondence. For attribute and relation changes, the invariant-difference representation helps distinguish true semantic differences from shared visual context. For distractor cases, the invariant semantic reference reduces responses to irrelevant viewpoint or illumination variations.

The qualitative results show that SIDFormer benefits from organizing change evidence before caption generation. Instead of directly decoding from raw before/after differences, the model first anchors visual correspondence with spatial cues, suppresses pseudo-change responses through invariant semantic reference, and then constructs compact difference-aware tokens for the decoder. This process helps SIDFormer generate captions that are more consistent with the actual semantic changes.