DiffTell: A High-Quality Dataset for Describing Image Manipulation Changes

Supplementary Material

Appendix

The supplementary material is composed as follows.

- Appendix A illustrates the motivation for designing the different captioning models for content authenticity.
- Appendix B presents a detailed description of the existing datasets in IDC.
- Appendix C gives the implementation details.
- Appendix D presents more baselines that are not included in the main paper.
- Appendix E presents the details using various instruction prompts.
- Appendix F presents the more results of SPICE and BertScore-F_1 as well as the human evaluation.
- Appendix G presents the out-of-distribution (OOD) results using Spot-the-Diff and CLEVR-DC datasets.
- Appendix H presents the zero-shot or few-shot performance on LLMs without being finetuned on IER testing set.
- Appendix I discusses more about the dataset collection.
- Appendix J presents more experiments and analysis about the automatic data filtering pipeline.
- Appendix K provides more details about PSBattle testing set.
- Appendix L provides more visual results, including how we filter the data, more successful cases, etc.

A. Motivation from Content Provenance and Authenticity

Figure 7 shows the motivation to design the different captioning models to ensure the authenticity of the content, which helps users decide where the image is from and what is modified over its original version.

B. Existing Datasets

The most commonly used datasets in the IDC task are CLEVR change [46], Spot-the-Diff [23], and Image Editing Request (IER) [59]. CLEVR change constitutes a sizable synthetic dataset characterized by moderate viewpoint variations. Spot-the-Diff is composed of pairs of frames extracted from video surveillance footage and the corresponding textual descriptions of visual changes. IER is crawled from the practical image editing requests from the Reddit channel, consisting of 3,939 pairs of real images, accompanied by 5,695 editing instructions. Each image pair in the training set is associated with one instruction. In contrast, each image pair is linked to three instructions for a more objective evaluation in the validation and testing sets. Because IER is collected from a real-world scenario, it covers more image difference categories, such as background change, text manipulation, and local object change. The definition of the image difference categories can be found in Section 3.2. Due to the single domain in CLEVR and Spot-the-Diff datasets, we mainly use IER in this work as the testing set, which aligns our scope to have a comprehensive, diverse, and practical dataset. For these two datasets, which are mainly about a single domain, we use them as out-of-distribution evaluation, which is given in Appendix G.

Category	Background	Text	Local object	Image style
Number of Images	117	53	277	223

Table 6. Statistics of each image difference category in the IER testing set.

C. Implementation Details

C.1. Training Details

For CLIP4IDC, We adopt the official implementation of CLIP4IDC. However, as it lacks the training script and the pretrained weights for IER, we reproduce the CLIP4IDC[4] model trained on IER exactly following its provided training hyper-parameter settings of the CLEVR dataset. For VARD-LSTM[5] and NCT[6], there is still no official implementation for IER and we reproduce them using the settings in CLEVR dataset. The pre-trained Biaffine Parse in NCT we use is from Diaparser[7]. For OpenFlamingo-3B, the vision encoder and language encoder are `ViT-L-14` and `anas-awadalla/mpt-1b-redpajama-200b`. The cross attention interval is 1. For LLaVA-Interleave-8B, the language model we use is `meta-llama/Meta-Llama-3-8B-Instruct`. For Fuyu-8B, we use `adept/fuyu-8b`. The training platform we use is 8 NVIDIA A100s with the 80GB GPU memory. The training epochs is 2 for the MLLMs. For the other hyper-parameters like learning rate, weight decay, batch size, please refer to Table 7.

D. The Performance of More Baselines

Besides the methods in the main text, we test more baselines including NCT [65] and VARD-LSTM [62] given in Table 8.

[4] https://github.com/sushizixin/CLIP4IDC
[5] https://github.com/tuyunbin/VARD
[6] https://github.com/tuyunbin/NCT
[7] https://github.com/Unipisa/diaparser

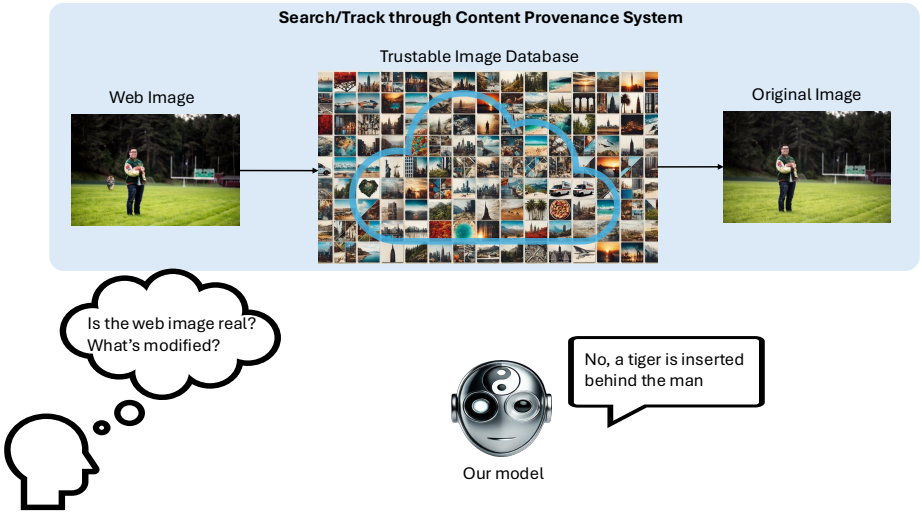


Figure 7. The overview of difference captioning for content provenance and authenticity. When the user asks if the web image is trustworthy, the content provenance system can search for its original version in the trustworthy image database if it exists. Our model is able to tell the user what is modified based on image difference captioning on the web image and the original one.

VL Model	Learning Rate	Epochs	Warmup Ratio	Weight Decay	Batch Sizes
OpenFlamingo-3B	5e-6	2	0.03	0.01	128
Fuyu-8B	1e-5	2	0.03	0.01	16
LLaVA-Interleave-8B	1e-5	2	0.03	0.0	16
Idefics3	5e-6	2	0.03	0.01	16
Qwen2-VL-7B-Instruct	5e-6	2	0.03	0.01	16

Table 7. The hyperparameters we use in this paper

Testing Set	Method	w/ *DiffTell*	BLEU@4	METEOR	CIDEr	ROUGE-L
IER	NCT	✗	1.64	7.97	7.47	19.40
		✓	**1.94**	**9.63**	**7.58**	**23.79**
IER	VARD-LSTM	✗	1.60	8.06	5.49	18.87
		✓	**1.71**	**8.54**	**6.02**	**20.08**
PSBattle	NCT	✗	2.78e-08	0.73	1.12	4.53
		✓	**1.65e-06**	**1.22**	**3.11**	**9.78**
PSBattle	VARD-LSTM	✗	1.49e-08	0.43	1.56	7.01
		✓	**7.46e-07**	**0.88**	**2.07**	**7.79**

Table 8. The comparison of the methods fine-tuned on image editing request (IER) training set with and without *DiffTell* using more baselines.

E. The Experiments with Diverse Prompts

In instruction tuning, incorporating diverse prompts enhances model robustness, making them more adaptable and better at generating accurate responses across varying contexts [11]. Initially, we use a uniform prompt "What is the difference between two images?" across all datasets and ask the model to provide an answer. To ablate this, we expand the prompt into nine different variations and compare the performance against the single-prompt approach, as shown in Table 9. The nine prompts we use are as follows. The model we use is OpenFlamingo-3B. As a complex vision-language task, it is more important for the model to understand two images, identify the difference and express the answer. Thus, to improve the vision encoder could be more useful.

- Please tell me the editing instruction of how to edit <|image|> to look like <|image|>.

Testing Set	w/ *DiffTell*	Diverse Prompt	BLEU@4	METEOR	CIDEr	ROUGE-L
IER	✗	✗	4.45	14.87	15.80	29.79
	✓	✗	**6.49**	**16.68**	21.04	**31.36**
	✓	✓	6.32	16.59	**23.88**	30.34
PSBattle	✗	✗	2.35e-04	2.33	**7.71**	**19.24**
	✓	✗	**2.12**	**6.60**	4.02	16.10
	✓	✓	1.77	6.45	4.48	16.46

Table 9. The results of performance on the IER testing set using the diverse prompts. The model we use is OpenFlamingo-3B.

Model	Idefics3-8B			Qwen2-VL-7B		
w/ *DiffTell*	Fluency (↑)	Correctness (↑)	Relevance (↑)	Fluency (↑)	Correctness (↑)	Relevance (↑)
✗	3.49	3.59	3.54	3.71	3.68	3.55
✓	**3.53**	**3.66**	**3.67**	**3.84**	**3.79**	**3.60**

Table 10. The results of human evaluation on Idefics3-8B and Qwen2-VL-7B. The data we use is the 50 pairs of images in the IER testing set. Each pair of images is labeled by 5 persons.

- Identify the transformations applied to <|image|> to achieve the appearance of <|image|>.
- Outline the steps required to edit <|image|> so that it matches the look of |image|>.
- Explain the edits necessary to convert <|image|> into <|image|>.
- What alterations were made to <|image|> to create <|image|>?
- Detail the changes from <|image|> to <|image|>.
- <|image|> is image1, <|image|> is image2, tell me what the change is between these two images.
- <|image|> is image1, <|image|> is image2, tell me what the change is from image1 to image2.

F. The Performance of Additional Evaluation Metrics

As mentioned in Section 4.1, we also evaluate the performance using SPICE and BertScore-F_1. The results are presented in Table 11.

For human evaluation, we randomly select 10% of the IER testing set (50 pairs of images) and let humans evaluate the output of Idefics3-8B and Qwen2-VL-7B trained with and without *DiffTell*. We use Amazon Mechanical Turk (MTurk) for this user study with each caption evaluated by 5 persons, ranging from 1 (worst) to 5 (best), resulting in 500 samples. The results shown in Table 10 demonstrate the effectiveness of *DiffTell*.

G. Out-of-Distribution Results

To evaluate the generalization capability of *DiffTell*, we test the model trained with and without *DiffTell* dataset on Spot-the-Diff and CLEVR-DC dataset without training on these two datasets. The results of Spot-the-Diff and CLEVR-DC using Qwen2-VL-7B are given in Table 12, showing that *DiffTell* can boost the performance of out-of-distribution (OOD) data, which is a good proof of its comprehensiveness.

H. Zero-shot/Few-shot Prompt Results

Investigating the potential of zero-shot learning is essential for methods utilizing LLM. For few-shot prompt testing, we randomly choose three examples from the IER training set. Performance results on the PSBattle dataset are not reported due to the absence of training data in that specific dataset. The detailed results can be found in Table 13. The few-shot prompt example is shown in Figure 8. The results show that image difference caption (IDC) is a hard task for the current LLMs, although they are trained on a huge amount of data. Even with few-shot prompt, the results are still not satisfying.

I. Dataset Collection Details

Image In-painting We use FireFly Generative Fill to in-paint the image. The inputs we can provide are the original image and the prompt for the generative model. There is no need for us to select the parameters. The illustration is given in Figure 11. We generate I_2 for COCO and MARIO-10M subsets in *DiffTell*.

Data Filtering The illustration of how the annotators filter the data is given in Figure 12, 13, and 14, which are for InstructPix2Pix, COCO, and MARIO-10M subsets, respectively. For InstructPix2Pix, the annotators filter whether the T_{I_1,I_2} matches (I_1, I_2) or whether the change reflects on I_1 and I_2 because (I_1, I_2, T_{I_1,I_2}) has already been provided.

Testing Set	IER Testing Set				PSBattle			
w/ *DiffTell*	✗	✓	✗	✓	✗	✓	✗	✓
Models	SPICE		BertScore-F_1		SPICE		BertScore-F_1	
OpenFlamingo-3B	9.31	**10.07**	87.13	**87.80**	3.77	**5.15**	85.50	**87.51**
Fuyu-8B	11.34	**16.07**	87.57	**88.95**	5.25	**6.60**	84.61	**87.56**
Llava-Interleave-7B	12.73	**16.86**	88.51	**89.20**	8.79	**8.88**	86.62	**87.67**
Idefics-8B	19.49	**22.64**	89.54	**90.49**	11.72	**13.37**	86.35	**87.45**
Qwen2-VL-7B	18.73	**21.86**	90.17	**90.48**	11.22	**11.78**	87.13	**87.64**

Table 11. The results of SPICE and BertScore-F_1 corresponding to Table 3 (main experiments) in the main paper.

Testing Set	w/ *DiffTell*	BLEU@4	METEOR	CIDEr	ROUGE-L	SPICE	BertScore-F_1
Spot-the-Diff	✗	2.51	4.88	4.12	11.45	6.51	85.34
	✓	**5.86**	**5.60**	**9.24**	**16.03**	**6.94**	**86.01**
CLEVR-DC	✗	0.99	2.53	2.27	6.71	17.07	85.71
	✓	**6.50**	**4.72**	**7.66**	**17.27**	**21.94**	**86.26**

Table 12. The results of out-of-distribution evaluation on Spot-the-Diff and CLEVR-DC datasets.

For COCO and MARIO-10M only providing I_1, the annotators filter whether the object or the text is successfully inpainted from I_1.

J. Extensive Experiments and Analysis on the Automatic Filtering

We further evaluate our automatic filter pipeline in the InstructPix2Pix dataset, using Qwen2-VL-7B as the feature extractor, in the same manner described in Section 4.4 in

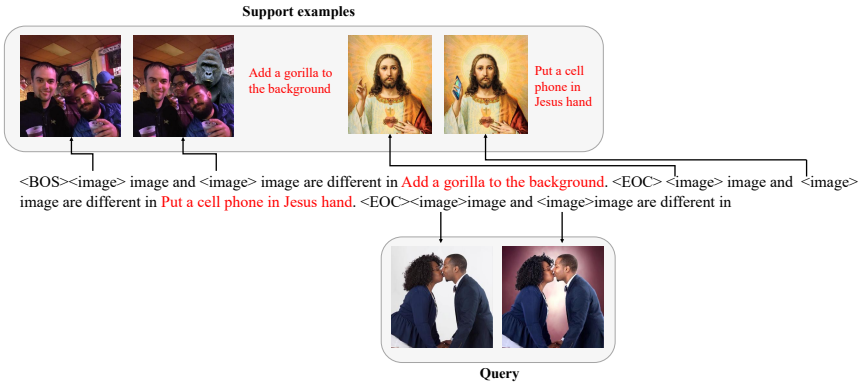


Figure 8. The example of how we construct the few-shot prompt.

Method	Few-shot	BLEU@4	METEOR	CIDEr	ROUGE-L
OpenFlamingo-3B	✗	1.18	8.07	8.72	16.63
	✓	0.84	7.64	4.09	17.54
OpenFlamingo-9B	✗	1.15	8.26	6.04	19.00
	✓	1.99	9.18	5.01	20.93

Table 13. The results of zero-shot or few-shot prompt results on the IER testing set. The few-shot prompt is the composition of 3 training examples from the training set.

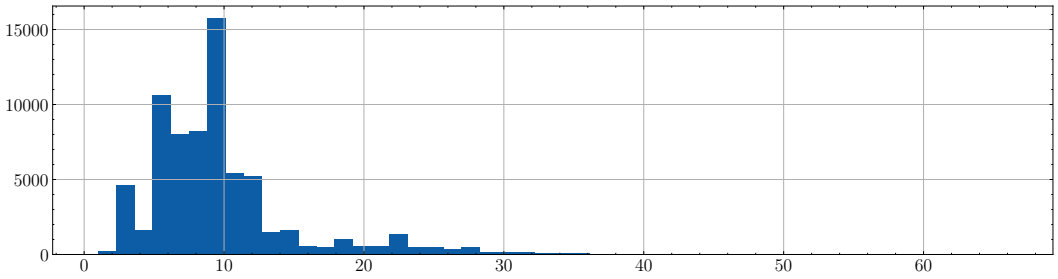


Figure 9. The difference description length distribution in *DiffTell*

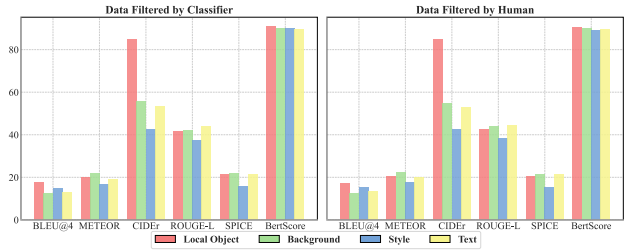


Figure 10. Comparison between data filtered by the classifier and human in terms of all the metrics.

the main paper. The accuracy of the data filter classifiers for InstructPix2Pix is 87.35%, respectively, indicating that the classifier has a satisfactory accuracy to keep clean data from a noisy dataset. The filtered data is used to train the Qwen2-VL-7B IDC model in Table 14. Due to the time limit, we compared the results with and without the automatically filtered 10K data from InstructPix2Pix in the last column of Table 14, indicating the **effectiveness** of the auto-filtered InstructPix2Pix data and the generalization of our automatic data filtering pipeline.

We further categorize the data into different types of differences and analyze the data filter accuracy on each type of difference, so we can analyze potential bias. We also conduct the **trade-off analysis** between the human and automatic data filtering using InstructPix2Pix data, with the result displayed in Figure 10. However, we do not observe an obvious performance trade-off in each difference type.

Training Set	B@4	M	C	R-L	S	B-F_1
IER	17.15	19.84	64.61	44.58	18.73	90.17
IER + 10K (R)	17.20	19.66	63.63	45.02	19.53	89.95
IER + 10K (C)	17.53	20.41	66.90	45.15	19.69	90.34
IER + 10K (H)	**17.97**	**20.67**	**68.02**	**45.25**	**19.70**	**90.46**

Table 14. The results of data filtering on the InstructPix2Pix dataset. R, C, and H refer to random data, data filtered by the classifier, and human filtering.

K. PSBattle Dataset

The PSBattle dataset is another practical dataset used in [8] that consists of images edited in Adobe Photoshop™ and is curated from the "Photoshopbattles" subreddit. We include this dataset only for the evaluation of out-of-domain data to test the generalizability of the models. This dataset comprises over 10,000 images, each paired with several modified variants generated according to editing instructions provided by users. In total, there are 102,208 variants created by 31,000 different artists. For our study, we randomly selected 100 image pairs, each accompanied by three captions obtained through crowd-sourced annotation on MTurk. The illustration of PSBattle dataset is shown in Figure 15.

L. More Visual Results

L.1. Failure Cases in Data Filtering

As mentioned in Section 3.3, we present the importance of the data filtering by showing more cases in InstructP2P, COCO, and MARIO-10M datasets in Figure 16, 17, and 18, respectively.

L.2. More Successful Cases

To better illustrate the improvement from *DiffTell*, we select another two prediction results in the IER testing set from the four categories, respectively, shown in Figure 19. The model we use is Qwen2-VL-7B.

(a) The image before in-painting

(b) The image after in-painting

Figure 11. We in-paint the image using Firefly Generative Fill in Photoshop. For each image, we provide the original image (I_1) and the corresponding mask. The mask is used to identify the selected area shown with the red arrow. We use a prompt to ask Firefly to in-paint the image and fit the background. Normally, the Firefly will return 3 to 4 in-painted images.

Instructions: Given an input image, the output image and the editting instruction. The meanings of the terms are as follows:

- Input Image: The original image we want to edit.
- Output Image: The image generated by AI model based on the input image.
- Editting Instruction: The instruction used to guide the AI model to generate the output image from the input image.

You are supposed to evaluate whether the ouput image is matched with the input image and the editting instruction. After carefully check the images and the instruction, you should select the quality score for the output image. Please check the Yes for the successful editting while No for the unacceptable editting.

input image output image

Editting Instruction: *make the creek dry*

Figure 12. The labeling illustration of InstructPix2Pix subsets. The two images are I_1 and I_2. T_{I_1,I_2} is given in **Editing Instruction**. The annotator is asked to identify whether the T_{I_1,I_2} matches (I_1, I_2) or whether the change reflects on I_1 and I_2 and give the answer "Yes" or "No". We keep those which are identified as "Yes".

Instructions: Given an input image, the input mask and a object-free image. The meanings of these 3 images are as follows:

- Input Image: The original image we want to remove the object.
- Input Mask: The region of the object generated by AI model. Ideally the mask should cover the object we want to remove.
- <object>-free Image: The image processed by AI model. **Ideally, there should not exist the <object> covered the mask and no extra element should be added. The <object> here is a placeholder which be will replaced by a specific object word.**

You are supposed to evaluate the object-free image, whether the object is fully removed without changing the original image content. After carefully compare the object-free image and the input image, you should select the quality score for how well the object is removed. We set 2 levels regarding the quality of the object-free image which are:

- Acceptable
- Unacceptable

The detailed criterion for the 2 categories and the corresponding example are given in the instrcution document.

input image | input mask | **airplane**-free image

Figure 13. The labeling illustration of COCO subsets. From left to right, the first, second, and third images are the original image (I_1), the input mask, and the in-painted image. We provide the input mask and object name to remind the annotator which area to focus on. The annotator selects "Acceptable" and "Unacceptable". We keep those which are identified as "Acceptable".

Instructions: Given an input image, the input mask and a text-free image. The meanings of these 3 images are as follows:

- Input Image: The original image we want to remove the text.
- Input Mask: The region of the text generated by AI model. Ideally the mask should cover all the text.
- Text-free Image: The image processed by AI model. **Ideally, there should not exist text and no extra element should be added.**

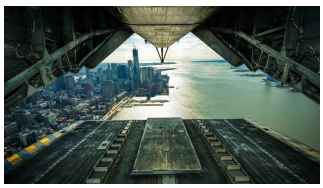
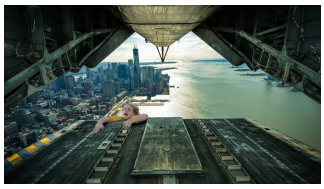
You are supposed to evaluate the text-free image, whether the text is fully removed without changing the original image content. After carefully compare the mask-free image and the input image, you should select the quality score for how well the text is removed. We set 2 levels regarding the quality of the object-free image which are:

- Acceptable
- Unacceptable

The detailed criterion for the 2 categories and the corresponding example are given in the instrcution document.

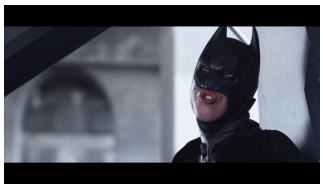
input image | input mask | text-free image

Figure 14. The labeling illustration of MARIO-10M subsets. From left to right, the first, second, and third images are the original image (I_1), the input mask, and the in-painted image. We provide the input mask and object name to remind the annotator which area to focus on. The annotator selects "Acceptable" and "Unacceptable". We keep those which are identified as "Acceptable".

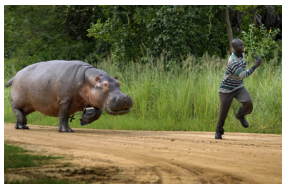



- hanging person added
- The right image has a person hanging off the end of the track with a horrified expression on his face.
- On the right, a man is clinging to the bomb bay door, about to fall. He is not there at all on the left.


- In the right picture the gun is visible
- Added Head hair in left eagle and cap and gun in the left one.
- Hawks are fighting each others in second one Hawk kept machine gun.




- A new face has been given to batman. I think it is the face of Will Ferral.
- The mask only covers part of the face and the man wears glasses now.
- Batman has been given a bushy head of hair and a large pair of glasses.

- The hippo is wearing a cross and holding a bible.
- The hippo is now carrying a bible and a crucifix necklace.
- The hippo is holding a bible and a crucifix in one of its hooves.

Figure 15. Four examples in PSBattle dataset.


make the mural of the Michelin Guide


make it 1920s


Add a river.


as a cartoon

Figure 16. Within the InstructP2P dataset, we have identified four sets of images, each composed of the original image, the altered image, and the corresponding instruction. All four of these image sets represent instances of failure. In the first pair of images, not only is the mural altered as per the instruction, but there are also changes to the face of the person in white and the text on the wall. The second pair exhibits subtle changes that are unrelated to the provided instruction. For the third pair, the images undergo significant alterations, including the addition of a river, surpassing the intended modifications. In the fourth pair, the changes between the two images fail to accurately reflect the given instruction. The InstructP2P dataset is characterized by a high noise ratio, leading to a low acceptance rate of 35.13% during manual filtering.

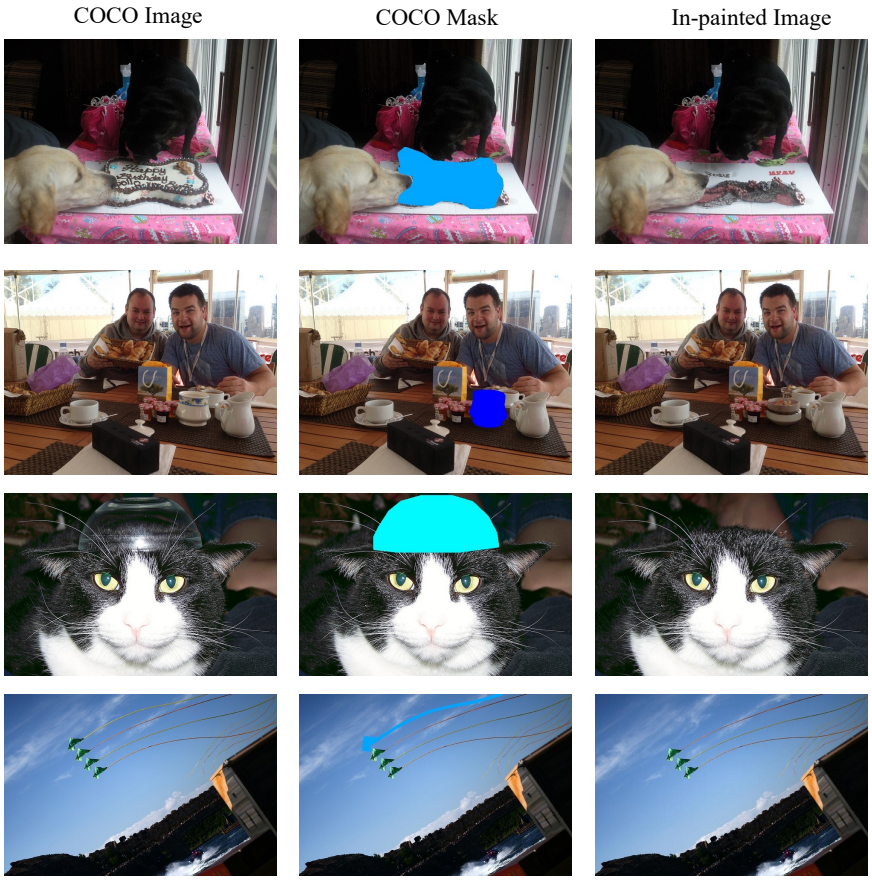


Figure 17. We choose four sets of images in the COCO dataset, each comprising the original image, the dilated mask, and the in-painted image. The initial two sets depict instances of failure, whereas the latter two sets showcase successful outcomes. The initial failure occurs when the mask fails to adequately cover the object, and the second failure is attributed to the inadvertent addition of another object despite the mask effectively covering the intended object. The labeling team is instructed to exclude images falling into *DiffTell*.

Figure 18. We select four image sets from the MAIRO-10M dataset, each including the original image, the dilated mask, and the in-painted image. All four of these cases have been deemed failures and subsequently excluded by the labeling team. The mask in the MARIO-10M dataset is not very precise. All of these 4 image sets show this issue. In the first image set, the text is not very clear, either. Besides the inadequate mask and addition objects which exist in the COCO dataset, another issue of MARIO-10M dataset is the existence of low-quality images.

Local Object

w/ DiffTell: Remove all the writing from the image.
W/o DiffTell: Remove all the text .
GT: Remove the background with all the signatures

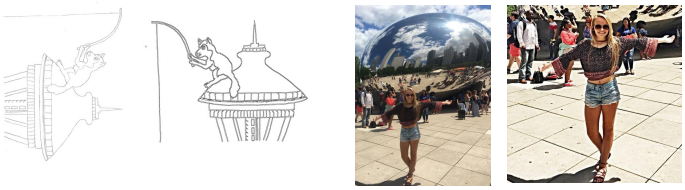
w/ DiffTell: Add masks to all three people.
W/o DiffTell: Add masks.
GT: Add mask to their faces.

Background

w/ DiffTell: Lighten the entire image.
W/o DiffTell: Remove the collar.
GT: lighten backgound

w/DiffTell: Crop the photo and make look clear.
W/o DiffTell: Remove the background .
GT: Make picture clear'

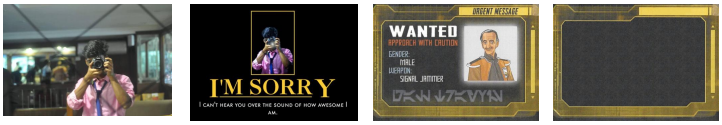
Image Style



w/ DiffTell: Rotate the image 90 degrees crop the image.
W/o DiffTell: Make the drawing more detailed and darker.
GT: Rotate image counter clockwise

w/ DiffTell: Crop the image .
W/o DiffTell: Remove the background.
GT: Crop this image and make it lighter

Text Editing



w/ DiffTell: change the background to black add the text IM SORRY CANT HEAR YOU OVER THE SOUND OF HOW AWESOME AM
w/o DiffTell: change the background to black add a yellow frame add text.
GT: add a black background to the whole image, crop the image, add the light on camera, add the whole image to the frame, add the text under the frame.

w/ DiffTell: remove the text URGENT MESSAGE WANTED APPROACH WITH CAUTION GENDER MALE WEAPON SIGNAL JAMMER..
W/o DiffTell: remove the wanted poster.
GT: Crop out the man with the camera and add him into a different background

Figure 19. More examples from the IER testing dataset regarding the four categories from Qwen2-VL-7B.