A Benchmark and Baseline for Language-Driven Image Editing

Jing Shi[1], Ning Xu[2], Trung Bui[2], Franck Dernoncourt[2], Zheng Wen[2], and Chenliang Xu[1]
[1]University of Rochester [2]Adobe Research
[1]{j.shi,chenliang.xu}@rochester.edu
[2]{nxu,bui,dernonco}@adobe.com zhengwen@alumni.stanford.edu

Abstract. Language-driven image editing can significantly save the laborious image editing work and be friendly to the photography novice. However, most similar work can only deal with a specific image domain or can only do global retouching. To solve this new task, we first present a new language-driven image editing dataset that supports both local and global editing with editing operation and mask annotations. Besides, we also propose a baseline method that fully utilizes the annotation to solve this problem. Our new method treats each editing operation as a submodule and can automatically predict operation parameters. Not only performing well on challenging user data, but such an approach is also highly interpretable. We believe our work, including both the benchmark and the baseline, will advance the image editing area towards a more general and free-form level.

1 Introduction

There are numerous reasons that people want to edit their photos, *e.g.*, remove tourists from wedding photos, improve saturation and contrast to make photos more beautiful, or replace background simply for fun. Therefore, image editing is very useful and important in people's everyday life. However, it is not a simple task for most people. One reason is that current mainstream photo editing softwares (*e.g.* Photoshop) could work only if users understand the concept of various editing operations such as hue, saturation, selection *etc.*, and know how to use them step by step. However, most novice users do not have such knowledge. Another reason is that most editing operations require some manual work, some of which could be very time-consuming. It is even more challenging when editing photos on mobile devices because people have to use their fingers while screen sizes are small.

In this paper, we propose *language-driven image editing* (LDIE) to make image editing easier for everybody. Specifically, to edit an image, a user only needs to provide a natural language request. Our new algorithm will automatically perform all the editing operations to produce the desired image without any manual intervention. The language request can be very detailed, including stepwise instructions such as "increase the brightness and reduce the contrast." But,

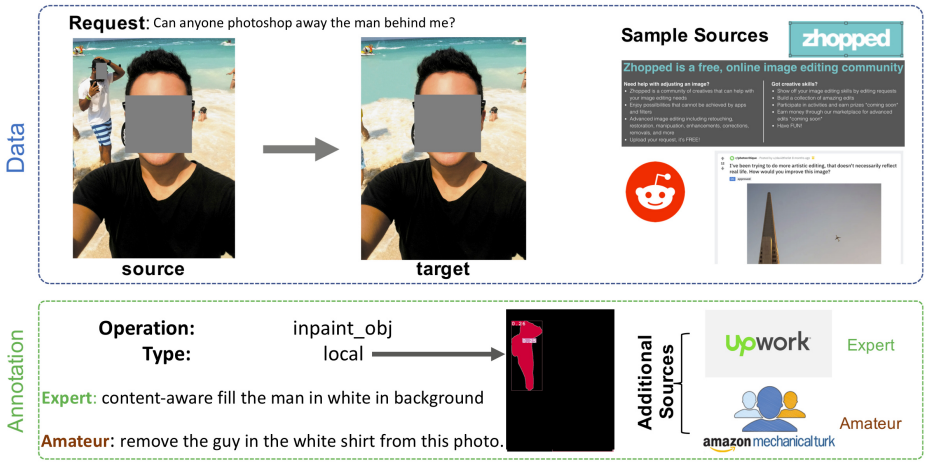

Fig. 1. One example in our newly collected Grounded Image Editing Request (GIER) dataset. Each sample is a triplet. We collected all samples from image-editing-request websites, *e.g.*, Zhopped and Reddit, and we augment language request data from both experts (Upwork) and the crowd-sourcing website (AMT).

it could also contain a certain level of vagueness (*e.g.* "make the sky bluer") or even very vague descriptions like "make the image more beautiful," which is particularly useful for novice users. One considerable advantage of the new task is that users no longer need to be involved in the tedious editing process (*e.g.*, determine editing operations and sequences, manual adjustment of parameters, masking *etc.*), which can be all accomplished by algorithms automatically.

There are a few previous studies that work on similar problems, but none of them can solve our new task. Many works [4,5,6,7] explore language-based manipulation for simple image contents such as birds, faces, or shoes. [8,9] only handle the image retouching operation and do not take any language inputs. Although being language-based, [10,11] only solve a single task in image editing (*e.g.*, retouching [10] or recoloring [11]), which are not extendable to other operations. PixelTone [12] solves the problem most similar to ours. However, it requires users to select editing regions manually and can only work for very detailed instructions.

Since no previous works directly solve our new task, we tackle it in two steps. We first collect a dataset named Grounded Image Editing Request (GIER) with 30k samples and 6k unique images, where each sample is a triplet, including one source image, one language request to edit the source image, and one target image which matches the editing request. Table. 1 illustrates the comparison of our datasets against the previous one and reflects the advantages of ours. All our image samples are real data collected from image-editing-request websites Zhopped.com and Reddit.com. We also augment language request data from both the crowd-sourcing website (AMT) and contracted experts (Upwork). We

Table 1. Comparison between GIER dataset and related existing datasets. *Size* is the number for unique images or image pairs (if paired). *User photo* and *user request* mean the image and request are general and are from real user. *Other annotation* is the annotation of editing mask or editing operation.

dataset	size	user photo	user request	paired image	other annotations
CUB [1]	11788	✗	✗	✗	✗
Oxford-102 [2]	8189	✗	✗	✗	✗
DeepFashion-Seq [3]	4820	✗	✗	✓	✗
CoDraw [4]	9993	✗	✗	✓	✗
i-CLEVER [4]	10000	✗	✗	✓	✗
IGER (Ours)	6179	✓	✓	✓	✓

believe our dataset will become an important benchmark in this domain, given its scale and high-quality annotation.

Next, we propose a baseline algorithm for the new task. Given a source image and a language request, our algorithm first predicts desired editing operations and image regions associated with each operation. Then, a modular network that comprises submodules of the predicted operations is automatically constructed to perform the actual editing work and produce the output image. The parameters of each operation are also automatically determined by the modular network. One advantage of our algorithm is its interpretability. At every step, it will produce some human-understandable outputs such as operations and parameters, which can be easily modified by users to improve the editing results. Besides, our method fully leverages all the dataset annotation, and each of its components helps check the quality of the dataset annotation.

We train our algorithm on our newly collected GIER dataset and evaluate it with ablation studies. Experimental results demonstrate the effectiveness of each component of our method. Thus, our method also sets a strong baseline result for this new task.

In summary, the contributions of this paper include:

- We propose a new LDIE task that handles both detailed and vague requests on natural images as well as both global and local editing operations.
- We collect the first large-scale dataset, which comprises all real user requests and images with high-quality annotations.
- We propose a baseline algorithm that is highly interpretable and works well on challenging user data.

The rest of the paper is organized as follows. In Sec. 2, we briefly introduce related work. Section. 3 describes our dataset and Sec. 4 describes the proposed algorithm in detail. Experimental results are given in Sec. 5, and finally, we conclude the paper in Sec. 6.

2 Related Work

Language-based image manipulation/generation. Many methods have been proposed recently for language-based image generation [13,14,15] and manipulation [4,5,6,7,11]. In this paper, we propose a new task, LDIE, which automatically performs a series of image editing operations given natural language requests. Methods for image manipulation/generation are dominated by variants of generative adversarial networks (GAN), and they change specific attributes of simple images, which usually only contain one primary object, e.g., faces, birds, or flowers. In contrast, our new task works on everyday image editing operations (*e.g.*, contrast, hue, inpainting *etc.*) applied to more complex user images from open-domain. A recent work [10] was proposed to edit images with language descriptions globally, and it collected a dataset that contains only global image retouching operations. In contrast, our new task handles both global and local editings, and our dataset comprises all real user requests, which cover diverse editing operations.

Image Editing. The task of image editing involves many subtasks such as object selection, denoising, shadow removal *etc.* Although many methods have been proposed for each subtask, how to combine the different methods of different subtasks to handle the more general image editing problem was seldom studied before. Laput *et al.* [12] proposed a rule-based approach which maps user phrases to specific image editing operations, and thus does not require any learning. However, this method is quite limited in that it requires each request sentence to describe exactly one editing operation. Besides, it cannot automatically determine operation parameters. There are several works [8,9,10] proposed for image retouching, which consider multiple global editing operations such as brightness, contrast, and hue. In [8,9], reinforcement learning is applied to learn the optimal parameters for each predefined operation. [10] leverages GANs to learn each operator as a convolutional kernel. Different from these two types of methods, we employ a modular network that was previously proposed for VQA [16,17,18], and learn optimal parameters for each predefined operator in a fully supervised manner.

Visual Grounding. Our algorithm needs to decide whether an operator is applied locally or globally and where the local area is if it is a local operator. Therefore, visual grounding is an essential component of our algorithm. However, previous visual grounding methods [19,20,10,21,22] are not directly applicable to our task due to the complexity of our language requests. For example, an expression for traditional visual grounding only contains the description of a single object. In contrast, our request contains not only object expression but also other information such as editing operators. Besides, each request may include multiple operators, and each could be a local one. Furthermore, each local region is not necessarily a single object. It could also be a group of objects or stuff (*e.g.*, "remove the five people on the grass"). Therefore the visual grounding problem is more challenging in our task.

3 The Grounded Image Editing Request (GIER) Dataset

In this section, we present how we collect a large-scale dataset called *Grounded Image Editing Request* (GIER) to support our new task.

3.1 Dataset Collection

Step 1: Preparation. First, we crawl user data from two image editing websites: Zhopped[1] and Reddit[2]. On the websites, amateur photographers post their images with editing requests, which are answered by Photoshop experts with edited images. Our crawled web data spans from the beginning of the two websites until 4/30/2019, resulting in 38k image pairs. Then, we construct a list of editing operations that cover the majority of the editing operations of the crawled data, which is shown in Tab. 2.

Step 2: Filtering and Operation Annotation. Although the crawled dataset is large, many samples are too challenging to include. There are mainly two challenges. First, some images contain local editing areas, which are hard to be grounded by the existing segmentation models due to the lack of training labels or other reasons. Second, some editing requests involve adding new objects or background into the original images, which cannot be easily handled by automatic methods.

Table 2. Statistics of all candidate operations. Each column represents the operation, the number of occurrence for each operation, the ratio of each operation over all operations , the ratio of images containing the operation over all images, and the ratio of local operation for each operation.

operation	#occur	opr%	img%	local%
brightness	3176	16.00	51.40	7.53
contrast	3118	15.70	50.46	4.84
saturation	2812	14.16	45.51	7.15
lightness	2164	10.90	35.02	6.93
hue	2059	10.37	33.32	11.56
remove object	1937	9.76	31.35	99.59
tint	1832	9.23	29.65	7.59
sharpen	842	4.24	13.63	6.18
remove bg	495	2.49	8.01	95.35
crop	405	2.04	6.55	23.95
deform object	227	1.14	3.67	17.18
de-noise	155	0.78	2.51	9.68
dehaze	133	0.67	2.15	11.28
gaussain blur	124	0.62	2.01	73.39
exposure	85	0.43	1.38	5.88
rotate	84	0.42	1.36	1.19
black&white	72	0.36	1.17	16.67
radial blur	65	0.33	1.05	83.08
flip image	23	0.12	0.37	0.00
facet filter	19	0.10	0.31	5.26
rotate object	12	0.06	0.19	41.67
find edges filter	10	0.05	0.16	0.00
flip object	7	0.04	0.11	85.71

To make our dataset more practical, we ask annotators to filter crawled samples belonging to the two challenging cases. To decide whether a local editing operation can be grounded or not, we preprocess each original image by applying the off-the-shelf panoptic segmentation model UPSNet [23] and let annotators check whether the edited areas belong to any pre-segmented regions.

After the filtering work, annotators are further asked to annotate the qualified samples with all possible operations from the operation list in Tab. 2 as well

[1] http://zhopped.com
[2] https://www.reddit.com/r/photoshoprequest

as the edited region of each operator. To get better-quality annotation, we hire Photoshop experts from Upwork to do the tasks. After the first-round annotation, we do another round of quality control to clean the low-quality annotations.
Step 3: Language Request Annotation. The crawled web data already contains one language request per sample. However, sometimes the original requests do not match the edited images well, which could cause problems for model training. Besides, we are interested in collecting diverse requests from different levels of photographers. Therefore we collect language requests from both the AMT and Upwork. AMT annotators usually have less knowledge about image editing, and Upwork annotators are Photoshop experts.

We present pairs of an original image and an edited image to AMT annotators without showing anything else. This is different from what we give to Upwork annotators, which contains additional information, including the original request as well as the annotation in step 2. To balance the data distribution, we collect three requests from AMT and two requests from Upwork. We also do another round of quality control to clean the bad annotations.

3.2 Data Statistics

The GIER dataset contains 6,179 unique image pairs. Each pair is annotated with five language requests, all possible editing operations, as well as their masks. The average number of operations per image pair is 3.21, and the maximum is 10. The distribution of each operation is show in Tab. 2. For language requests, the average word length is 8.61; the vocabulary size is 2,275 (after post-processing).

Our newly-collected GIER dataset is highly valuable for LDIE task. First, all data are from real users' editing requests so that they genuinely reflect a large portion of the needs for image editing. Second, we collect language requests from diverse users, which are helpful in making methods trained on our dataset practical for real applications. Third, our dataset is annotated with many different types of ground truth, which makes learning different types of methods possible.

4 A Baseline for Language-Driven Image Editing

We define the task of LDIE as follows. Given an original image and a user request, a method needs to produce an output image that matches the editing request. The closer the output image is to the target image, the better the method is. Contrast from the prevalent GAN-based methods, we propose the baseline model that can edit by sequentially applying interpretable editing operations, requiring the comprehension of both language request and visual context. Since most operations are resolution-independent, our model can also keep the image resolution same as the input. The main body of our model is an operation modular network (Sec. 4.3) shown in our model pipeline (Fig. 2). It stacks multiple editing operations in order and predicts best parameters. Since the layout of operations is discrete variable which is hard to optimize only given the target image, we resort to a supervisely trained operation classifier (Sec. 4.1) to predict

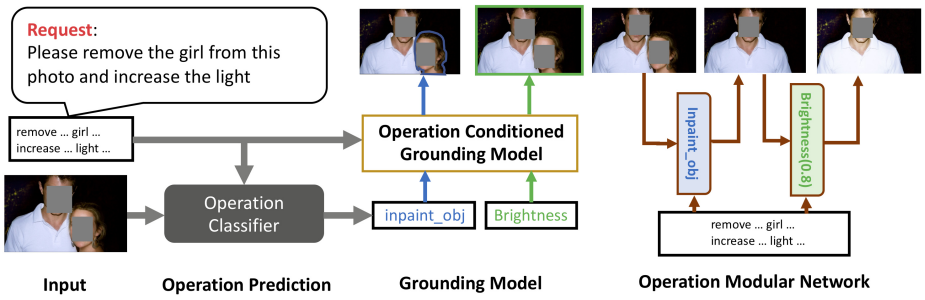

Fig. 2. An overview of the model's pipeline. The input image and request are sent to a multi-label classifier to predict operations. Then, the operation conditioned grounding model takes in the image, the request, and operations and outputs the grounding mask for each operation. Finally, the operation modules are cascaded to form the operation modular network, and each step outputs an interpretable intermediate result.

needed editing operations and arrange them in a fixed order. Moreover, every editing operation requires a mask to specify where to edit, which is obtained by the operation conditioned grounding model (Sec. 4.2). Although our model is not completely end-to-end trained, it is a valuable initial attempt to address such task in such a compositional and interpretable way.

4.1 Operation Prediction

Since samples in GIER are annotated with ground truth operations, we train a multi-label classification network to predict operations. In Tab. 2, there are 23 operations, while some of them have too few training examples. Therefore, we pick the top nine operations (`brightness` and `lightness` are merged as one operation due to their similarity) as our final classification labels. They cover 90.36% of total operations and are representative of users' editing habits. Both input image and language request are input to the classifier, owning to many unspecific requests which require the perception of the input image. The image is embedded by ResNet18 [24], and the language request is embedded by a bi-directional LSTM [25]. The two features are then concatenated and passed into several fully connected layers to get the final prediction. This model is trained with the multi-label cross-entropy loss.

4.2 Operation Conditioned Grounding Model

In our task, the language of request may contain multiple types of operation-based groundings (*e.g.*, "please remove the girl from this photo and increase the light" in Fig. 2) and each grounding may contain multiple, even disconnected regions (*e.g.*, "remove all pedestrians from the image"). Given such uniqueness of our task, the previous visual grounding methods are not directly applicable. However, they certainly serve as a good starting point. In this section, we will

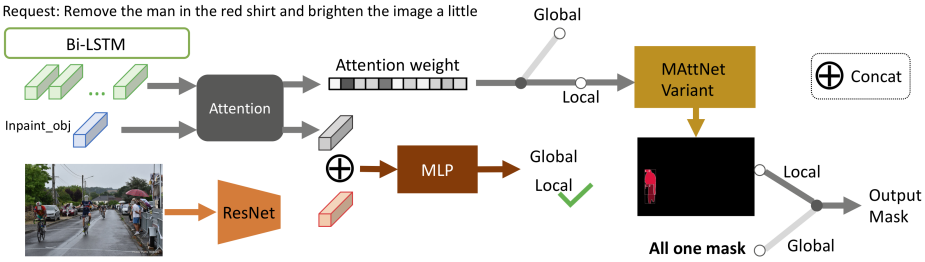

Fig. 3. Network structure of the operation-conditioned grounding model. The operation attends to its related part in the request. And the MLP binary classifier can judge if the operation is local or global. If local, the MAttNet variant will ground the operation related description into mask, otherwise output the all one mask.

first review a state-of-the-art visual grounding model, MattNet [26], and then show step-by-step how to turn it into a proper grounding model for its use in our task taking into consideration of the operation input and multi-region output.

The Grounding Problem and MattNet. Given a request Q, an operation O, and an image I, the goal is to localize the area of the image to be edited by the operator. We formulate it as a retrieving problem by extracting the region proposals $R = \{R_i\}$ from the image and choosing one or more of the region proposals to make up the target area.

The basic MattNet comprises a language attention network and three visual modules—subject, location, and relationship. The language attention network takes the query Q as input and outputs the modular phrase embeddings for each module $[q^{subj}, q^{loc}, q^{rel}]$ and their attention weights $\{w_{subj}, w_{loc}, w_{rel}\}$. Each module individually calculates the matching score between its query embedding and the corresponding visual feature. Then the matching scores from three modules are weighted averaged by $\{w_{subj}, w_{loc}, w_{rel}\}$. Finally, the ranking loss for positive query-region pair (Q_i, R_i) and negative pair (Q_i, R_j) is:

$$\begin{aligned} L_{rank} = \Sigma_i(& \max(0, \Delta + s(Q_i, R_j) - s(Q_i, R_i)) + \\ & \max(0, \Delta + s(Q_j, R_i) - s(Q_i, R_i))), \end{aligned} \tag{1}$$

where $s(x, y)$ denotes the matching score between query x and region y, and Δ denotes the positive margin.

Operation Conditioned Language Attention. We extend the language attention network of MattNet. The reason for choosing MattNet is that the editing request frequently describes the objects in the subject-location-relationship format. The request Q of length T represented by word vectors $\{e_t\}_{t=1}^{T}$ is encoded by a Bi-LSTM and yields the hidden vectors $\{h_t\}_{t=1}^{T}$. The operation word embedding is o. The operation finds its corresponding noun phrase in the request by using attention. Therefore, the attention weights from the operation to all

the request tokens are:

$$\alpha_t^{(o)} = \frac{\exp(\langle o, h_t \rangle)}{\sum_{k=1}^{T} \exp(\langle o, h_k \rangle)}, \tag{2}$$

where $\langle \cdot, \cdot \rangle$ denotes inner product, and the superscript (o) indicates the specific attention for operation o. We keep trainable vectors f_m, where $m \in \{\text{subj, loc, rel}\}$, from MattNet to compute the attention weights for each of three visual modules:

$$a_{m,t} = \frac{\exp(\langle f_m, h_t \rangle)}{\sum_{k=1}^{T} \exp(\langle f_m, h_k \rangle)}. \tag{3}$$

Then, we can compute an operation conditioned attention and thus, obtain operation conditioned modular phrase embedding:

$$\hat{a}_{m,t} = \frac{\alpha_t a_{m,t}}{\sum_{k=1}^{T} \alpha_k a_{m,k}}, \qquad q_m^{(o)} = \sum_{t=1}^{T} \hat{a}_{m,t}^{(o)} e_t. \tag{4}$$

The other parts of the language attention network remain the same. For the visual modules, we keep the location and relationship modules unchanged. For the subject module, we remove the attribute prediction branch because the template parser [19] is not suitable for our editing request.

Multiple Object Grounding. Since we formulate the task as a retrieving problem, we set a threshold for the matching score to determine multiple grounding objects. If all objects are under the threshold, the top-1 object will be selected. However, an operation might be grounded to the whole image, which requires the model to retrieve all the candidates. To remedy such a problem, we add an extra binary classifier to tell if the operation is local or global, given the context of image and request. The structure is presented in Fig. 3.

Since GIER dataset provides ground truth instance masks, the operation-conditioned grounding model is trained with the ranking loss as Eq. 1.

4.3 Operation Modular Network

After the set of possible operations, along with their masks, are predicted, our method constructs a Operation Modular Network (OMN) to perform the actual editing work. The OMN is composed of submodules, each of which represents a predefined editing operation. Each submodule takes as an input image or the previously edited image, the language request and the mask, and produces an output image. See Fig. 4 for an illustration. The training objective for OMN is learning to predict the best parameter for each operation from the supervised of the target image. Next, we first describe the implementation of each submodule, then the way how we create the modular network, and finally, the loss functions.

Submodule Implementation. We create one submodule for each chosen operation. Among them, six are implemented by differentiable filters which are also resolution independent. Specifically, `brightness` and `saturation` are implemented by scaling the HSV channels. `sharpness` is achieved by adding the

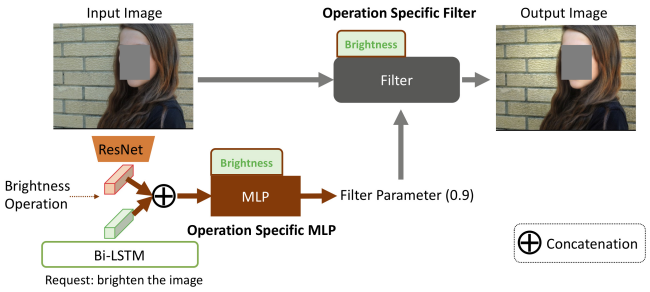

Fig. 4. The structure of a submodule network for the `brightness` operation.

image spatial gradient. `contrast`, `hue`, and `tint` are implemented the same as [8]. For `remove_bg` we simply implement `remove_bg` as changing the masked area to white given our sample distribution, which is non-differentiable. And `inpaint_obj` is implemented by a differentiable neural inpainting model EdgeConnect [27]. Refer to Supplement C for more implementation details.

Except `remove_bg` and `inpaint_obj`, the other operations also require some input parameters, which are automatically predicted by their submodules. Specifically, the request and image features are concatenated and sent to an MLP to predict the parameter. The filter takes in the parameter and mask and yields the output image. Each operation has its individual MLP parameters.

Modular Network Creation. The modular network is created by linking together all predicted operations. However, `remove_bg` is non-differentiable thus would blocked the gradient backpropagation. And `inpaint_obj` is a large network that is computational expensive for gradient. Luckily, these two submodules do not have any parameters to learn. Therefore, we always put them in front of the chain if they exist.

Loss Function. The L1 loss is applied between the final output image I_K and the target image I_{gt} to drive the output image to be similar to the target image:

$$\text{Loss}_{l1} = |I_K - I_{gt}|, \tag{5}$$

where K denotes the number of predicted operations, *i.e.* the length of the submodule chain.

However, only using the supervision at the final step might not guarantee that the intermediate images are adequately learned. Hence, we also propose to use a step-wise triplet loss to enforce the intermediate image to be more similar to the target image than its previous step:

$$\text{Loss}_{tri} = \frac{1}{K}\sum_{k=0}^{K-1} \max(|I_{k+1} - I_{gt}| - |I_k - I_{gt}| + \Delta, 0), \tag{6}$$

where Δ is a positive margin. It resembles triplet loss by regarding I_{gt} as anchor, I_k as negative sample and I_{k+1} as positive. Note that we should block the gradient of the term $|I_k - I_{gt}|$ to prevent from enlarging the distance between I_{gt} and I_k. Hence final loss is $\text{Loss} = \text{Loss}_{l1} + \lambda\text{Loss}_{tri}$, with balanced weight λ.

Table 3. The F1 score and ROC-AUC score for operation prediction.

	F1					ROC
threshold	0.3	0.4	0.5	0.6	0.7	
val	.7658	.7699	.7620	.7402	.7026	.9111
test	.7686	.7841	7759	.7535	.7172	.9153

Table 4. The operation type classification accuracy

	Accuracy					ROC
threshold	0.1	0.3	0.5	0.7	0.9	
val	.9328	.9328	.9328	.9328	.9328	.8915
test	.9377	.9387	.9397	.9397	.9397	.8969

Table 5. The grounding results

	F1					IoU					ROC
threshold	0.15	0.20	0.25	0.30	0.35	0.15	0.20	0.25	0.30	0.35	
val	.6950	.7286	7412	.7280	.6700	.5788	.6328	.6519	.6254	.5439	.8857
test	.6953	.7432	.7626	.7350	.6380	.5682	.6296	.6578	.6203	.5161	.9186

5 Experiment

5.1 Experiment Setup

Dataset. We train and evaluate our model in our GIER dataset. The dataset is split into training, validation and testing subset with ratio 8:1:1, resulting in 4934/618/618 image pairs, respectively.

Metrics. For operation prediction, it is a multi-label classification task, hence we evaluate it using F1 score and ROC-AUC. For the operation conditioned grounding, we evaluate two sub-tasks: operation binary classification (local or global) evaluated by accuracy, and the local operation grounding evaluated by F1 score, ROC-AUC and IoU. Since the local operation grounding is formulated as a multi-object retrieving task, F1 score and ROC-AUC are reasonably set as the metrics. Moreover, the selected multiple objects can make up a whole image-level mask, so we also evaluated the mask quality using IoU score computed between the grounded mask and the ground truth mask. To evaluate the final output image, we adopt L1 distance between the predicted image and the target image, where the pixel are normalized from 0 to 1. However, since the request could have many suitable editing, we further conduct human study to get more comprehensive evaluation. The implementation detail is in Supplement. D.

5.2 Results: Operation Prediction

The result for operation prediction is shown in Tab. 3. We evaluate F1 score under different confidence thresholds and observe that the validation and test set has the similar trend and achieve best performance at threshold 0.4. And the ROC score also indicate a good performance on operation prediction and can support the later task well. Its visualization can be found in Supplement. B.1.

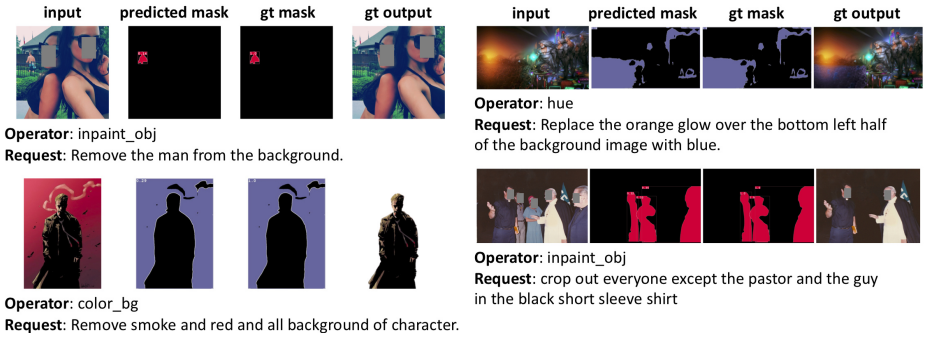

Fig. 5. Visualization for the operation conditioned grounding

Table 6. The comparison between a GAN-based method with our method. The arrow indicates the trend for better performance.

	L1↓	User Rating ↑	User Interact ↑
Target	-	3.60	-
Random Edit	0.1639	-	-
Pix2pixAug [10]	**0.1033**	2.68	13.5%
Our method (UB)	0.0893	-	-
Our method	0.1071	**3.20**	**86.5%**

Table 7. Ablation study 1 and 2 with V and L representing vision and language

Study	Metric	L	V+L
1	ROC	0.9182	0.9153
2	ROC	0.9804	0.8969
	Acc@0.5	0.9508	0.9397

5.3 Results: Operation Conditioned Grounding

For operation type classification, the accuracy is listed in Tab. 4. For local operation grounding, the quantitative result is in Tab. 5. F1 score and IoU are evaluated under various confidence thresholds with the same trend, and both attain peak value at threshold 0.25. The ROC score is 0.8857 and 0.9186 for validation and test set, respectively. The evaluation result indicating a good start for the operation modular network. Fig. 5 shows the qualitative grounding results for local operations. In many cases the request is to remove distraction persons in the background, such as the first and last row in Fig. 5, requiring the grounding model to distinguish the high-level semantic of foreground and background. Also, the cartoon figures images make the grounding even more challenging. The visualization of the operation attention is in Supplement. B.2.

5.4 Results: Language Driven Image Editing

The main quantitative results are shown in Tab. 6 with L1 and two user evaluation metrics. The comparison methods are described as follows. *Pix2pixAug* is a GAN-based model following the language-augmented pix2pix model in [10]. *Random Edit* is sequentially apply random editing operations with random parameters in random number of steps. *Our method UB* is the performance upper

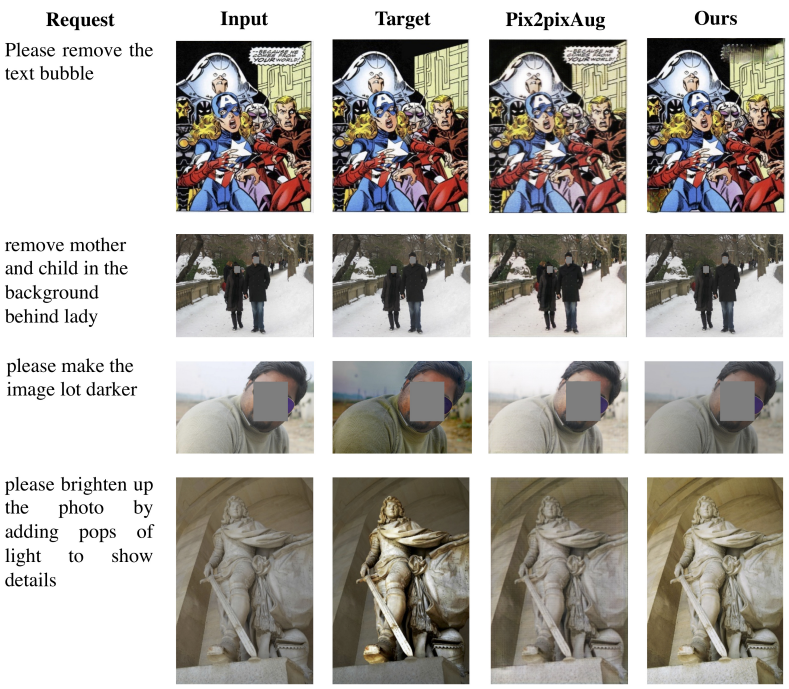

Fig. 6. The visual comparison between our method and a GAN-based method. The first two rows are local editing, our method can correctly remove the designated object, even for text, while pix2pixAug cannot do such local editing. And for the last two rows, our method has more salient editing than pix2pixAug.

bound for OMN where the ground truth operations and masks are given as input. *Our method* is our full model where operations and masks are predicted. Experiments show that Pix2pixAug has slightly better L1 score, but the user rating and user interactive ratio (detailed in Supplement. A.1) strongly indicates that our method is more perceptually appealing to humans, and of more advantageous for human-interactive editing. Also, the performance gap between our method and its upper bound indicates that better operation and mask prediction can bring a large performance gain. Figure 6 demonstrates that our method has better awareness for local editing and more salient editing effect than Pix2pixAug. More visualization of our edited images is in Supplement. A.2.

5.5 Ablation Study

Study 1: To investigate the importance of the visual information, we compare the operation prediction performance by using 1) only language feature (L) and 2) concatenation of vision and language feature (V+L). The result is listed in Tab. 7. We find that pure language feature is comparable with both vision and

Table 8. The comparison between OMN with triplet loss and without triplet loss.

	w/ Triplet	w/o Triplet
L1	0.0893	0.0925

Table 9. The comparison between OMN with fixed and random operation order.

	Fixed order	Random order
L1	0.0893	0.0875

language feature, indicating that the language information itself usually already contains rich context for operation selection.

Study 2: Also for the grounding task, we compare the global or local classification with or without visual information provided . The comparison is drawn in Tab. 7. It reveals that purely using language feature is a better way to decide if an operation is local or global. We suspect the reason is that if the operation is described with a location or object phrase, then such operation is of high possibility to be a local operation, so the visual information may not be so helpful compared with language.

Study 3: we explore the effectiveness of the triplet loss applied on each generation step in upper bound setting. Table. 8 shows that with the triplet loss the OMN achieves better performance, demonstrating its positive effect.

Study 4: The effect of the operation order is evaluated in Tab. 9 in upper bound setting. We compare the models trained and test in fixed order and random order, and the result is slightly better for random order than fixed order.

6 Conclusion and Future Direction

In this paper, we propose the LDIE task along with a new GIER dataset which supports both local and global editing and provides object masks and operation annotations. We design a baseline modular network to parse the whole request and execute the operation step-by-step, leading to an interpretable editing process. To handle the unique challenges of visual grounding in this new task, we propose the operation conditioned grounding model extending the MattNet to consider operation input and multi-region output.

Currently our model uses the intermediate operation and mask as supervision to facilitate the modeling and in turn evaluate the annotation quality. However such intermediate operation annotation might contain human bias and how to learn the model that only supervised by target image can be further explored. For evaluation metrics, LDIE task should also evaluate whether the edit is applied to the correct region specified by language. We evaluated this according to the grounding performance, which rely on the intermediate mask ground truth. However, a more general evaluation only depending on target image can be proposed. Finally, more editing operations can be added to the model.

Acknowledgement: This work was partly supported by Adobe Research, NSF 1741472 and 1813709. The article solely reflects the opinions and conclusions of its authors but not the funding agents.

References

1. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset. (2011) 3
2. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing, IEEE (2008) 722–729 3
3. Zhu, S., Urtasun, R., Fidler, S., Lin, D., Change Loy, C.: Be your own prada: Fashion synthesis with structural coherence. In: ICCV. (2017) 3
4. El-Nouby, A., Sharma, S., Schulz, H., Hjelm, D., Asri, L.E., Kahou, S.E., Bengio, Y., Taylor, G.W.: Tell, draw, and repeat: Generating and modifying images based on continual linguistic instruction. In: ICCV. (2019) 2, 3, 4
5. Nam, S., Kim, Y., Kim, S.J.: Text-adaptive generative adversarial networks: manipulating images with natural language. In: NeurIPS. (2018) 2, 4
6. Cheng, Y., Gan, Z., Li, Y., Liu, J., Gao, J.: Sequential attention gan for interactive image editing via dialogue. arXiv preprint arXiv:1812.08352 (2018) 2, 4
7. Shinagawa, S., Yoshino, K., Sakti, S., Suzuki, Y., Nakamura, S.: Interactive image manipulation with natural language instruction commands. arXiv preprint arXiv:1802.08645 (2018) 2, 4
8. Hu, Y., He, H., Xu, C., Wang, B., Lin, S.: Exposure: A white-box photo post-processing framework. ACM Transactions on Graphics (TOG) **37** (2018) 26 2, 4, 10
9. Park, J., Lee, J.Y., Yoo, D., So Kweon, I.: Distort-and-recover: Color enhancement using deep reinforcement learning. In: CVPR. (2018) 2, 4
10. Wang, H., Williams, J.D., Kang, S.: Learning to globally edit images with textual description. arXiv preprint arXiv:1810.05786 (2018) 2, 4, 12
11. Chen, J., Shen, Y., Gao, J., Liu, J., Liu, X.: Language-based image editing with recurrent attentive models. In: CVPR. (2018) 2, 4
12. Laput, G.P., Dontcheva, M., Wilensky, G., Chang, W., Agarwala, A., Linder, J., Adar, E.: Pixeltone: A multimodal interface for image editing. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM (2013) 2, 4
13. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis. arXiv preprint arXiv:1605.05396 (2016) 4
14. Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., He, X.: Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In: CVPR. (2018) 4
15. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N.: Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: ICCV. (2017) 4
16. Andreas, J., Rohrbach, M., Darrell, T., Klein, D.: Neural module networks. In: CVPR. (2016) 4
17. Hu, R., Andreas, J., Rohrbach, M., Darrell, T., Saenko, K.: Learning to reason: End-to-end module networks for visual question answering. In: ICCV. (2017) 4
18. Mao, J., Gan, C., Kohli, P., Tenenbaum, J.B., Wu, J.: The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. ICLR (2019) 4
19. Kazemzadeh, S., Ordonez, V., Matten, M., Berg, T.: Referitgame: Referring to objects in photographs of natural scenes. In: EMNLP. (2014) 4, 9

20. Hu, R., Xu, H., Rohrbach, M., Feng, J., Saenko, K., Darrell, T.: Natural language object retrieval. In: CVPR. (2016) 4
21. Yang, Z., Gong, B., Wang, L., Huang, W., Yu, D., Luo, J.: A fast and accurate one-stage approach to visual grounding. In: ICCV. (2019) 4
22. Liu, X., Wang, Z., Shao, J., Wang, X., Li, H.: Improving referring expression grounding with cross-modal attention-guided erasing. In: CVPR. (2019) 4
23. Kirillov, A., He, K., Girshick, R., Rother, C., Dollár, P.: Panoptic segmentation. In: CVPR. (2019) 5
24. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016) 7
25. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE transactions on Signal Processing **45** (1997) 2673–2681 7
26. Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M., Berg, T.L.: Mattnet: Modular attention network for referring expression comprehension. In: CVPR. (2018) 8
27. Nazeri, K., Ng, E., Joseph, T., Qureshi, F., Ebrahimi, M.: Edgeconnect: Generative image inpainting with adversarial edge learning. arXiv preprint arXiv:1901.00212 (2019) 10