Learning by Planning: Language-Guided Global Image Editing

Jing Shi[1] Ning Xu[2] Yihang Xu[1] Trung Bui[2] Franck Dernoncourt[2] Chenliang Xu[1]
[1]University of Rochester [2]Adobe Research
[1]{j.shi,chenliang.xu}@rochester.edu [1]yxu74@u.rochester.edu [2]{nxu,bui,dernonco}@adobe.com

Abstract

Recently, language-guided global image editing draws increasing attention with growing application potentials. However, previous GAN-based methods are not only confined to domain-specific, low-resolution data but also lacking in interpretability. To overcome the collective difficulties, we develop a text-to-operation *model to map the vague editing language request into a series of editing operations, e.g., change contrast, brightness, and saturation. Each operation is interpretable and differentiable. Furthermore, the only supervision in the task is the target image, which is insufficient for a stable training of sequential decisions. Hence, we propose a novel operation planning algorithm to generate possible editing sequences from the target image as pseudo ground truth. Comparison experiments on the newly collected MA5k-Req dataset and GIER dataset show the advantages of our methods. Code is available at* https://github.com/jshi31/T2ONet.

1. Introduction

Image editing is ubiquitous in our daily life, especially when posting photos on social media such as Instagram or Facebook. However, editing images using professional software like PhotoShop requires background knowledge for image processing and is time-consuming for the novices who want to quickly edit the image following their intention and post to show around. Furthermore, as phones and tablets becoming users' major mobile terminal, people prefer to take and edit photos on mobile devices, making it even more troublesome to edit and select regions on the small screen. Hence, automatic image editing guided by the user's voice input (*e.g.* Siri, Cortana) can significantly alleviate such problems. We research global image editing via language: given a source image and a language editing request, generate a new image transformed under this request, as firstly proposed in [34]. Such a task is challenging because the model has to not only understand the language but also edit the image with high fidelity. Rule-based methods [22, 21] transfer the language request into sentence templates and further map the templates into a sequence of executable editing operations. However, they require additional language annotations and suffer from unspecific editing requests. [30] directly maps the language to operations with the capability to accept the vague editing request, yet still need the operation annotation for training. A more prevalent track is the GAN-based method [34], which models the visual and textual information by inserting the image and language features into a neural network generator that directly outputs the edited image. However, GAN-based models lack the interpretability about how an image was edited through a sequence of common editing operations (*e.g.* tone, brightness). Thus, they fail to allow users to modify the editing results interactively. Moreover, GANs struggle with high-resolution images and is data-hungry.

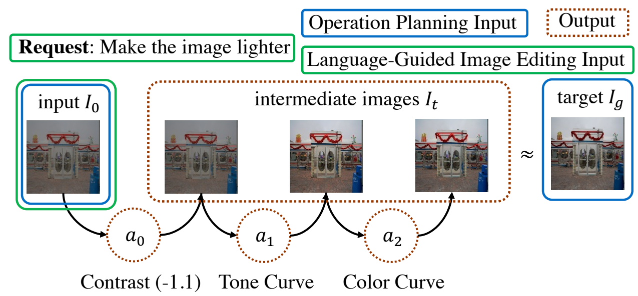


Figure 1. Language-Guided Global Image Editing: given the input image I_0 and the request, we predict a sequence of actions a_t to edit the image progressively with a series of intermediate images I_t generated. And the final edited image is our output, which should accord with the request. Operation Planning: the input image I_0 and target image I_g are given, and we plan a sequence of action to make the final edited image reach the target image I_g.

To provide an interpretable yet practical method for language-guided global image editing, in this paper, we propose a Text-to-Operation Network (T2ONet). The network sequentially selects the best operations from a set of predefined everyday editing operations to edit the image progressively according to the language's comprehension and the visual editing feedback. As the operations are resolution-independent, such method will not deteriorate the image resolution. Fig. 1 shows the process of mimicking human experts for professional photo editing and opens the possibility for human-computer interactions in future work.

One crucial difficulty for training our model is the lack of supervision information for editing sequences—we do not have access to intermediate editing operations and their parameters. The only available supervision is the input image's tuple, the target image, and the language editing request. One possible solution is to train our model by Reinforcement Learning (RL). For example, the model can try different editing sequences and get rewards by comparing the edited images to the target images. However, it is well-known that RL is highly sensitive to hyper-parameters and hard to train when the action space is large (*e.g.* high-dimensional continuous action). On the other hand, it is demanding yet infeasible to collect annotations for all intermediate operations and their parameters in practice. Therefore, a novel training schema is expected to solve our task. To overcome this difficulty, we devise a weakly-supervised method to generate pseudo operation supervision. Inspired from the classical forward search planning [29], we propose an operation-planning algorithm to search the sequence of operations with their parameters that can transform the input image into the target image, as shown in Fig. 1. It works as an inverse engineering method to recover the editing procedure, given only the input and the edited images. Such searched operations and parameters serve as pseudo supervision for our T2ONet. Also, as the target image is used as the pixel-level supervision, we prove its equivalence to RL. Besides, we show the potential of the planning algorithm to be extended to local editing and used to edit a new image directly.

In summary, our contributions are fourfold. First, we propose T2ONet to predict interpretable editing operations for language-guided global image editing dynamically. Second, we create an operation planning algorithm to obtain the operation and parameter sequence from the input and target images, where the planned sequences help train T2ONet effectively. Third, a large-scale language-guided global image editing dataset MA5k-Req is collected. Fourth, we reveal the connection between pixel supervision and RL, demonstrating the superiority of our weakly-supervised method compared with RL and GAN-based methods on AM5k-Req and GIER [30] datasets through both quantitative and qualitative experimental results.

2. Related Work

Language-based image editing. Language-based image editing tasks can be categorized into one-turn and multi-turn editing. In one-turn editing, the editing is usually done in one step with a single sentence [6, 26, 24, 18]. Dong *et al*. [6] proposed a GAN-based encoder-decoder structure to address the problem. Nam *et al*. [26] leverage the similar generator structure but use a text-adaptive discriminator to guide the generator in the more detailed word-level signal. However, both [6, 26] simply use concatenation to fuse the textual and visual modalities. Mao *et al*. [24] proposes the bilinear residual layer to merge two modalities to explore second-order correlation. Li *et al*. [18] further introduces a text-image affine combination module to select text-relevant area for automatic editing and use the detail correction module to refine the attributes and contents. However, the above works are built on the "black box" GAN model and inherit its limitations. Shi *et al*. [30] introduces a new language-guided image editing (LDIE) task that edits by using interpretable editing operations, but its training requires the annotation of the operation.

For multi-turn editing, the editing request is given iteratively in a dialogue, and the edit should take place before the next request comes [7, 4]. However, only toy datasets are proposed for this task.

Our task belongs to a variant of one-turn editing that focuses on global image editing, which is proposed in [34], which also uses a GAN-based method by augmenting the image-to-image structure [14] with language input. Different from all the above, our method can edit with complex language and image via understandable editing operations without the need for operation annotations

Image editing with reinforcement learning. To enable interpretable editing, [13] introduces a reinforcement learning (RL) framework with known editing operations for automatic image retouching trained from unpaired images. However, it cannot be controlled by language requests. **Task planning.** Task planning aims at scheduling a sequence of task-level actions from the initial state to the target state. Most related literature focuses on the pre-defined planning domain through symbolic representation [25, 8, 17]. Our *operation planning* is reminiscent of task planning[29]. However, it is hard to use symbolic representation in our case because of high-dimensional states and continuous action space.

Modular networks. The modular networks are widely adopted in VQA [1, 11, 15, 10, 36, 23] and Visual Grounding [12, 20, 37]. In the VQA task, the question is parsed into a structured program, and each function in the program is a modular network that works specifically for a sub-task. The reasoning procedure thus becomes the execution of the program. However, the parser has discrete output, and it is usually trained with program semi-supervision [11, 15] or with only the final supervision in an RL fashion [23]. LDIE task has a similar setting that only the target image is given as supervision, but we facilitate our model training by our planning algorithm.

3. Method

We achieve the language-guided image editing by mapping the editing request into a sequence of editing operations, conditioned on both input image and language. We propose T2ONet to achieve such mapping (Sec. 3.3). The

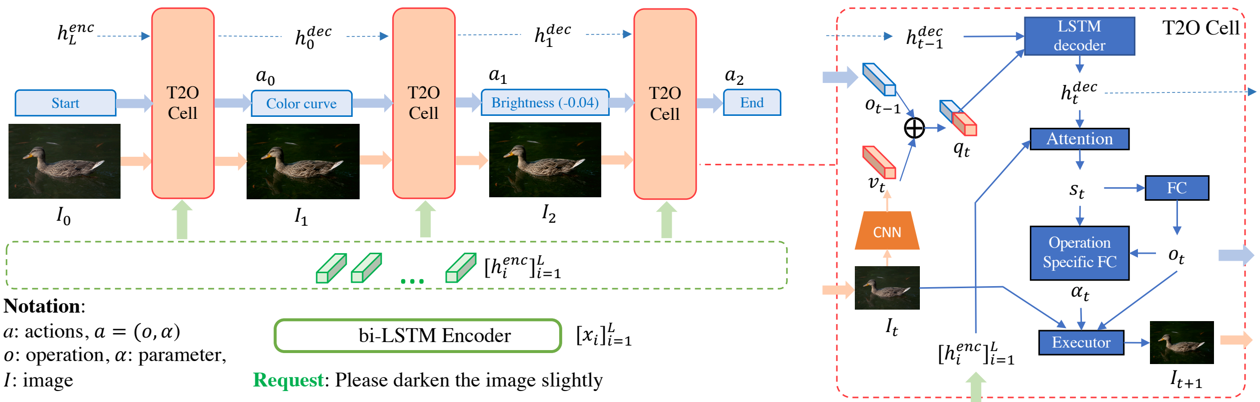


Figure 2. Structure of the T2ONet. An LSTM encoder embeds the request, and the T2O-Cell progressively decodes the input image and request into action and image series. At each step t, the T2O-Cell generates the next action a_t and image I_{t+1} based on previous operation o_{t-1}, hidden state h_{t-1}^{dec}, and image I_t.

critical difficulty is that we only have the target image's supervision but no supervision of the sequence. To tackle this difficulty, we introduce the idea of planning into the modeling to obtain a feasible operation sequence as the pseudo ground truth (Sec. 3.4). Finally, we talk about the training process (Sec. 3.5) and the connection to RL (Sec. 3.6).

3.1. Problem Formulation

Starting with an input image I_0 and a language request Q, the goal is to predict an output image similar to the target image I_g. In contrast to the GAN-based model, which outputs the edited image in one step, we formulate the editing problem through a sequential prediction of action sequence $\{a_t\}_{t=0}^T$ with length $T+1$ to edit the input image following the language request. Applying a_t to I_t leads to I_{t+1}, and the final action a_T is END action that will not produce new image, as shown in Fig. 2. In this way, the model generates a sequence of images $\{I_t\}_{t=1}^T$, where I_T is the final output or target image. An action is defined as $a = (o, \alpha)$, where o is the choice of discrete editing operations, and α is the continuous parameter of the operation.

3.2. Operation Implementation

We adopt six operations: *brightness*, *saturation*, *contrast*, *sharpness*, *tone*, and *color*. Among them, *brightness* and *saturation* is implemented by scaling H and S channels in the HSV space [9], controlled by a single re-scaling parameter. *Sharpness* is implemented by augmenting the image with spatial gradients, controlled by a single parameter. *Contrast* is also a single-parameter operation and implemented following [13]. *Tone* is controlled by eight parameters that construct a pixel value mapping curve, following [13]. Finally, *color* is similar to *tone* but is implemented with three curves that operate on each of RGB channels, each controlled by eight parameters. The details of the operation implementation are in the Appx. H.

3.3. The Text-to-Operation Network (T2ONet)

We propose the T2ONet to map the language request and the input image to a sequence of actions, which optimizes the joint action distribution, where each new action is predicted based on its past actions and intermediate images:

$$P(\{a_t\}_{t=0}^T | I_0, Q) = P(a_0 | I_0, Q) \times \prod_{t=1}^{T} P(a_t | \{a_\tau\}_{\tau=0}^{t-1}, \{I_\tau\}_{\tau=0}^{t}, Q). \quad (1)$$

We denote state s_t as the condensed representation of $\left(\{a_\tau\}_{\tau=0}^{t-1}, \{I_\tau\}_{\tau=0}^{t}, Q\right)$, then the objective is transformed to: $P(\{a_t\}_{t=0}^T | s_0) = \prod_{t=0}^T P(a_t|s_t)$. To realize the policy function $P(a_t|s_t)$, we adopt an Encoder-Decoder LSTM architecture [5], shown in Fig. 2. The request $Q = \{x_i\}_{i=1}^L$ is encoded using a bi-directional LSTM upon the GloVe word embeddings [28] into a series of hidden states $\{h_i^{enc}\}_{i=1}^L$ and the final cell state m_L^{enc}. Then, an LSTM decoder is represented as $h_{t+1}^{dec}, m_{t+1}^{dec} = f(h_t^{dec}, m_t^{dec}, q_t)$, where $q_t = \text{concat}(\text{Embedding}(o_t); v_t)$. o_t, h_t^{dec}, and m_t^{dec} are the predicted operation, the hidden state, and the cell state at the t-th step, respectively (we omit m_t^{dec} in Fig. 2 for simplicity). Similar to word embedding, each operation is embedded into a feature vector through a learnable operation embedding layer. $v_t = \text{CNN}(I_t)$ denotes the image embedding via CNN at the t-th step. Then, the attention mechanism [2] is applied to better comprehend the language request $\beta_{ti} = \frac{\exp\left((h_t^{dec})^\mathrm{T} h_i^{enc}\right)}{\sum_{i'=1}^L \exp\left((h_t^{dec})^\mathrm{T} h_{i'}^{enc}\right)}$, $c_t = \sum_{i=1} \beta_{ti} h_i^{enc}$, $s_t = \tanh(W_c[c_t; h_t^{dec}])$. The state vector s_t is now the mixed feature of past images, operations, and the language request. Since the parameter α is dependent on the operation o, we further decompose the policy function as $P(a_t|s_t) = P(o_t, \alpha_t|s_t) = P(o_t|s_t)P(\alpha_t|o_t, s_t)$, where $P(o_t|s_t)$ is obtained through a Fully-Connected (FC) layer

Algorithm 1: Operation Planning

```
Input: I_0, I_g, max operation step N, threshold ε,
       beamsize B, operation set O
p=[I_0]
cost(I) = ||I − I_g||_1
for t in 1 : N do
    q ← [ ]
    for I ∈ p do
        for o ∈ O do
            α* = arg min_α cost(o(I, α))
            I* ← o(I, α*)
            q ← q ∪ I*
        end
    end
    q ← Sort(q), sortkey = cost(I*)
    p = q[: B]
    for I ∈ p do
        if cost(I) < ε then
            Break All Loop
        end
    end
end
{o_t}, {α_t}, {I_t} ← Backtracking(p)
return {o_t}, {α_t}, {I_t}
```

to predict the operation o_t, which is expressed as:

$$P(o_t|s_t) = \text{softmax}(W_o s_t + b_o). \tag{2}$$

For parameter prediction $P(\alpha_t|o_t, s_t)$, different operations can have different parameter dimensions. Therefore, we create an operation-specific FC layer for each operation to calculate: $\alpha_t = W_\alpha^{(o)} s_t + b_\alpha^{(o)}$, where superscription (o) is the indicator of the specific FC layer for operation o. Hence, $P(\alpha_t|o_t, s_t)$ is modeled as a Gaussian distribution $\mathcal{N}(\alpha_t; \mu_{\alpha_t}, \sigma_\alpha)$:

$$P(\alpha_t|o_t, s_t) = \mathcal{N}(\alpha_t; W_\alpha^{(o_t)} s_t + b_\alpha^{(o_t)}, \sigma_\alpha). \tag{3}$$

Finally, the executor will apply the operation o_t and its parameter α_t to the image I_t to obtain the new image I_{t+1}. The process from I_t to I_{t+1} will repeat until the operation is predicted as the "END" token.

3.4. Operation Planning

To provide stronger supervision for training policy function, we introduce the operation planning algorithm that can reverse engineer high-quality action sequences from only the input and target images. Concretely, given the input image I_0 and the target image I_g, plan an action sequence $\{a_t\}_0^T$ to transform I_0 into I_g. This task is similar to the classical planning problem [8], and we solve it with the idea of forward-search. Algorithm 1 shows the operation planning process. We define the planning model with action a, image I as state, and state-transition function $I' = o(I, \alpha)$, where o is the operation. The state transition function takes image I and parameter α as input and outputs a new image. The goal is to make the final image I_T similar to I_g as within an error ϵ, specified by the L1 distance $||I_T - I_g||_1 < \epsilon$. To reduce redundant edits, we restrict each operation to be only used once and limit the maximum edit step to N.

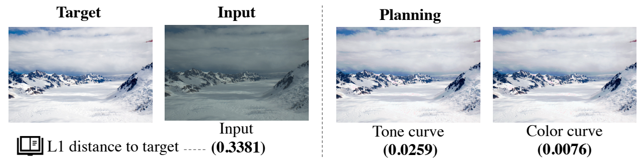


Figure 3. Visualization of the operation planning trajectory. The number The L1 distance is monotonically decreasing and can recover highly similar result to the target.

In algorithm 1, we wrap the goal into a cost function and try to minimize the cost during each step. However, the action a includes both discrete operation o and continuous parameter α, which could be high-dimensional with extremely large searching space. To make computing efficient, we only loop over all the discrete operation candidates, but as the operation is chosen, we optimize the parameter to minimize the cost function. Such optimization could significantly reduce the searching space for parameters. Since all operations here are differentiable, the optimization process could be 0th-, 1st-, and 2nd-order, *e.g.*, Nelder-Mead [27], Adam [16], and Newton's method, respectively. At each step t, the algorithm visits every image in the image candidate list of beam size B, and for each image, the algorithm enumerates the operation list of size $|\mathcal{O}|$. Since it has at most N steps, the maximum time complexity for operation planning is $O(NB|\mathcal{O}|)$. In practice, we constraint the planning for unrepeated operations. Fig. 3 shows one trajectory of our planned sequence, as it stops at the second step since the cost is lower than $\epsilon = 0.01$. Different operation sets and orders are studied in Sec. 4.5. We further show two potential extensions of the operation planning algorithm.

Extension1: Planning through a discriminator. The $\text{cost}(I)$ is not limited to $||I_T - I_g||_1$, but can be the image quality score yield by a pretrained discriminator D without dependence of the target image. Then our operation planning can directly edit new images (see Sec. 4.6 for details).

Extension2: Planning for local editing. Although our paper focuses on global editing, the operation planning can be extended to planning local editing by searching the region masks with an additional loop, detailed in Sec. 4.6.

3.5. Training

The planning algorithm 1 creates pseudo ground truth operation $\{o_t^*\}_{t=0}^T$ and parameter sequence $\{\alpha_t^*\}_{t=0}^{T-1}$ to supervise our model. The operation is optimized by minimizing the cross-entropy loss (XE):

$$\mathcal{L}_o = -\sum_{t=0}^{T} \log(P(o_t^*|s_t)). \tag{4}$$

Maximizing the log-likelihood for Eq. 3 equals to applying MSE loss:

$$\mathcal{L}_\alpha = \sum_{t=0}^{T-1} ||\alpha_t - \alpha_t^*||_2^2. \tag{5}$$

Additionally, to utilize the target image supervision, we apply the image loss as final L1 loss as:

$$\mathcal{L}_{L1} = ||I_T - I_g||_1. \tag{6}$$

The ablation study (Appx. A.1) proves the L1 loss is critical for better performance. Although teacher forcing technique is a common training strategy in sequence-to-sequence model [32], where the target token is passed as the next input to the decoder, teacher forcing does not work for $\mathcal{L}_{L1}$ since the intermediate pseudo-GT input blocks the gradient. Therefore we train $\mathcal{L}_{L1}$ in a non-teacher forcing fashion and $\mathcal{L}_o, \mathcal{L}_\alpha$ in the teacher forcing fashion, alternatively. Our final loss is $\mathcal{L} = \mathcal{L}_o + \mathcal{L}_\alpha + \mathcal{L}_{L1}$.

More request-sensitive output. The model is expected to be request-sensitive: produce diversified edits following different requests, rather than simply improve the image quality regardless of the requests. To improve the request-sensitivity, we propose to sample the parameter α_t from $\mathcal{N}(\alpha_t; \mu_{\alpha_t}, \sigma_\alpha)$ in Eq. (3) to train the image loss. In our default setting, $\sigma_\alpha = 0$, *i.e.* $\alpha_t = \mu_{\alpha_t}$. Our motivation is that sampling the parameter will produce stochastic editing results, preventing the model from falling into one same editing pattern or shortcuts regardless of the language. Also, there exist multiple reasonable edits for one request, so the $\mathcal{L}_{L1}$ still guarantees the stochastic output images to be reasonable. We observe that increasing σ_α leads to higher request-sensitivity (see Sec. 4.5). In fact, the next section will discuss the above training scheme for image loss with a close relation with RL.

3.6. Equivalence of Image Loss and DPG

To bridge the equivalence, we adapt an RL baseline from [13]. Due to space limitations, the detailed introduction of the baseline is in Appx. B.1, here we focus on the training for parameter α with RL and its connection to image loss. Let the reward be $r_t = \text{cost}(I_{t-1}) - \text{cost}(I_t)$, policy $\pi_o = P(o|s)$ in Eq. (2), $\pi_\alpha = \mathcal{N}(\alpha; \mu_\alpha, \sigma_\alpha)$, the accumulated reward defined as $G_t = \sum_{\tau=0}^{T-t} \gamma^\tau r_{t+\tau}$ ($\gamma = 1$ as [13]), the goal is to optimize the objective $J(\pi) = \mathbb{E}_{(I_0,Q)\sim P(\mathcal{D}), o\sim\pi_o \alpha\sim\pi_\alpha} G_1$. The continuous policy π_α is optimized by Deterministic Policy Gradient algorithm (DPG) [31]. Different from the common setting [31, 13] where the Q function is approximated with a neural network to make it differentiable to action, we approximate Q as G since our G_{t+1} is already differentiable to α_t, resulting in the DPG for each episode as

$$\nabla_{\theta_\alpha} J(\pi) = \sum_{\mathbb{E}\ t=0}^{T-1} \nabla_{\alpha_t} G_{t+1} \nabla_{\theta_\alpha} \alpha_t. \tag{7}$$

Now, we show the equivalence between image loss and DPG using the following theorem:

Theorem 1. *The DPG for α in Eq. (7) can be rewritten as*

$$\nabla_{\theta_\alpha} J(\pi) = -\frac{\partial \text{cost}(I_T)}{\partial \theta_\alpha}. \tag{8}$$

Proof. See Appx. B.2 □

Theorem 1 provides a new perspective that minimizing the $\mathcal{L}_{L1}$ for the final image in T2ONet is actually equivalent to optimizing the model with deterministic policy gradient at each step.

4. Experiments

4.1. Datasets

MA5k-Req. To push the research edge forward, we create a large-scale language-guided global image editing dataset. We annotate language editing requests based on MIT-Adobe 5k dataset [3], where each source image has five different edits by five Photoshop experts, leading to a new dataset called MA5k-Req. 4,950 unique source images are selected, and each of the five edits is annotated with one language request, leading to 24,750 source-target-language triplets. See Appx. J.1 for data collection details. We split the dataset as 17,325 (70%) for training, 2,475 (10%) for validation, and 4,950 (20%) for testing. After filtering the words occurring less than 2 times, the vocabulary size is 918. Note that [34] also similarly creates a dataset with 1884 triplets for this task, but unfortunately, it has not been released and is 10 times smaller than ours.

GIER. Recently, GIER dataset [30] is introduced with both global and local editing. We only select the global editing samples, leading to a total of 4,721 unique image pairs, where each is annotated with around 5 language requests, resulting in 23,171 triplets. we splits them as 18,571 (80%) for training, 2,404 (10%) for validation, and 2,196 (10%) for testing. After filtering the words occurring less than 3 times, the vocabulary size is 2,102.

4.2. Evaluation Metrics

Similar to the L2 distance used in [34], we use L1 distance, Structural Similarity Index (SSIM), and Fréchet Inception Distance (FID) for evaluation. L1 distance directly measures the averaged pixel absolute difference between the generated image and ground truth image as the pixel

	MA5k-Req					GIER				
	L1 ↓	SSIM↑	FID↓	$\sigma_{\times 10^2}$↑	User↑	L1↓	SSIM↑	FID↓	$\sigma_{\times 10^2}$↑	User↑
Target	-	-	-	-	3.5053	-	-	-	-	3.6331
Input	0.1190	0.7992	12.3714	-	-	0.1079	0.8048	49.6229	-	-
Bilinear GAN [24]	0.1559	0.4988	102.1330	0.8031	1.9468	0.1918	0.4395	214.7331	1.2164	1.7988
Pix2pixAug [34]	0.0928	0.7938	14.5538	0.5401	3.0957	0.1255	0.7293	74.7761	**1.2251**	2.5148
SISGAN [6]	0.0979	0.7938	30.9877	0.1659	2.8032	0.1180	0.7300	140.1495	0.0198	2.1243
TAGAN [26]	0.1335	0.5429	43.9463	1.5552	2.5691	0.1202	0.5777	112.4168	0.6073	2.4970
GeNeVa [7]	0.0933	0.7772	33.7366	0.6091	3.0851	0.1093	0.7492	87.0128	0.5732	2.7278
RL	0.1007	0.8283	7.4896	**1.6175**	3.1968	0.2286	0.3832	132.1785	0.3978	1.8462
T2ONet	**0.0784**	**0.8459**	**6.7571**	0.7190	**3.3830**	**0.0997**	**0.8160**	**49.2049**	0.6226	**2.8994**

Table 1. Quantitative results on two test sets. $\sigma_{\times 10^2}$ means that the image variance has been scaled up 100 times.

range is normalized to 0-1. SSIM measures image similarity through luminance, contrast, and structure. FID measures the Fréchet distance between two Gaussians fitted to feature representations of the Inception network over the generated image set and ground truth image set. To further exam the model's language-sensitivity, we propose the image variance σ to measure the diversity of the generated image conditioned on different requests. Similar to [19], we apply 10 different language requests (see Appx. I) to the same input image and output 10 different images. Then we compute the variance over the 10 images of all pixels and take the average overall spatial locations and color channels. Finally, we take the average of the average variance over the entire test set. The variance can only measure the diversity of generated images in different language conditions but cannot directly tell the editing quality. So we still resort to user study to further measure the editing quality.

User study setting. We randomly select 250 samples from the two datasets, respectively, with each sample evaluated twice. The user will see the input image and request and blindly evaluate the images predicted by different methods as well as the target image. Each user rates a score from 1 (worst) to 5 (best) based on the edited image quality (fidelity and aesthetics) and whether the edit accords with the request. We collect the user rating through Amazon Mechanical Turk (AMT), involving 42 workers.

4.3. Implementation Details

For operation planning, we set the maximum step $N = 6$, tolerance $\epsilon = 0.01$, and constraint that one operation is only used once. We adopt Nelder-Mead [27] for parameter optimization. The model is optimized by Adam [16] with learning rate 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$. More details are elaborated in Appx. G.

4.4. Main Results

Operation planning. The set 5 in Tab. 2 shows the averaged L1 distance of the planning result is 0.0136, which is around only 3.5-pixel value error towards target images, with pixel range 0-255. Fig. 3 shows the operation planning can achieve the visually indistinguishable output compared with the target. So we are confident to use the planned action sequence as a good pseudo ground truth.

Comparison methods.

- *Input*: the evaluation between input and target image.
- *Bilinear GAN* [24], *SISGAN* [6], *TAGAN* [26]: these three methods are trained by learning the mapping between the caption and image without image pairs. Since there is not image caption in our task but the paired image and request, we drop the procedure of image-caption matching learning but adapt them with the L1 loss between input and target images.
- *Pix2pixAug* [34]: the pix2pix model [14] augmented with language used in [34].
- *GeNeVa* [7]: a GAN-based dialogue guided image editing method. We use it for single-step generation.
- *RL*: out RL baseline introduced in Sec. 3.6.

We also compared with ManiGAN [18], but its output is very blurred as it is not designed for our task, and its network lacks the skip connection structure to keep the resolution. So we just show its visualization in Appx. F.1.

Result analysis. The qualitative and quantitative comparison are in Fig. 4 and Tab. 1, respectively. However, the results of BilinearGAN, TAGAN are bad, and their visual results have been omitted. For interested readers please refer to Appx. F.1. Fig. 4 shows that SISGAN has obvious artifacts, Pix2pixAug, and GeNeVa have less salient editing than ours, the RL tends to be overexposed in Fivek-Req and does not work well on GIER. Our T2ONet generates more aesthetics and realistic images, which are most similar to targets. The much worse performance of BilinearGAN, TAGAN, SISGAN might because their task is different from ours and their model ability is limited for complex images. Tab. 1 demonstrates that our T2ONet achieves the best performance on visual similarity metrics L1, SSIM, and FID, but not the σ. Firstly, σ can measure the editing diversity, as in Fig. 6; however, the σ and visual similarity metric are usually a trade-off, as shown in Sec. 4.5. So although RL has the highest σ under MA5k-Req, it sacrifices L1 much more, and its visual results indicate that it tends

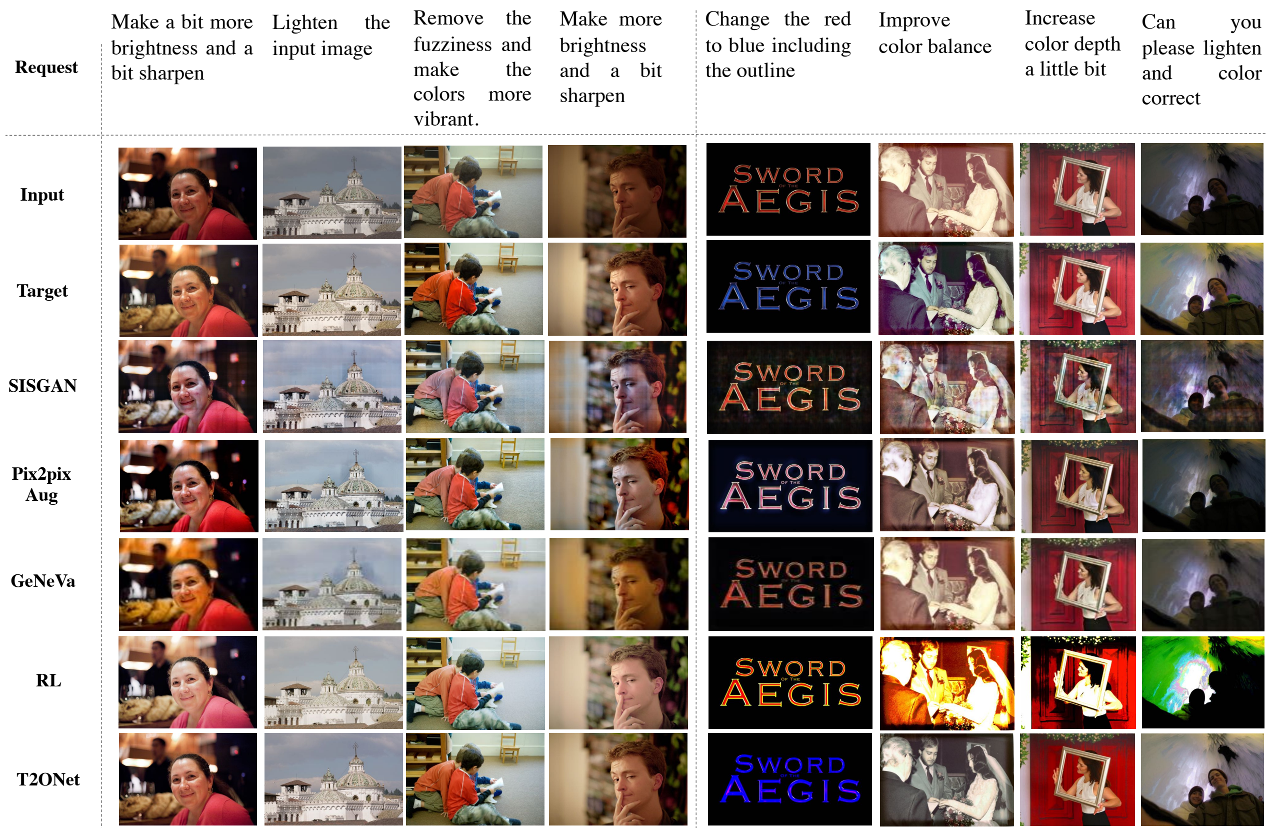


Figure 4. Visualization for comparison of our method T2ONet with other methods on MA5k-Req (left) and GIER (right).

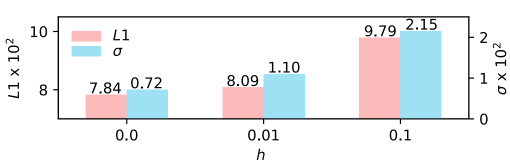


Figure 5. L1 and variance trade-off by training with different parameter sample variance on the MA5k-Req test set.

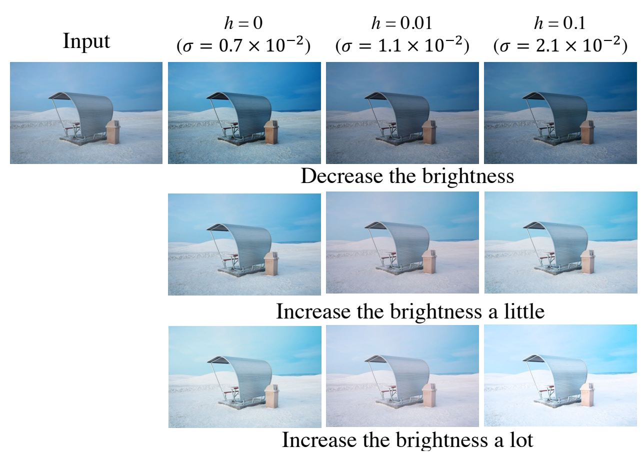


Figure 6. The same input edited with different language by models trained with different h. Image variance σ for the whole test data is also shown as a reference. The model trained with larger h has more diversified output.

to be overexposed. Second, the σ might be dominated by noisy random artifacts, *e.g.*, BilinearGAN in Fig. 4. Therefore, we resort to user ratings for best judgment, which indicates our method is the most perceptually welcomed.

Dataset Comparison. Tab. 1 also reflects the difference between the two datasets. Since GIER has a smaller data size and contains more complex editing requests, GIER is more challenging than MA5k-Req, which is verified by the fact that the gap of the user rating between target and T2ONet is much larger on GIER than on MA5k-Req.

Advantage over GAN. GAN-based methods also suffer from high-resolution input and can be jeopardized by artifacts. However, our T2ONet is resolution-independent without artifacts (see Appx. E.1).

Advantage over RL. With the more challenging GIER dataset, it makes RL harder to explore the positive-rewarded actions and fail. However, T2ONet still works well on GIER with the help of the pseudo action ground truth from operation planning. We further show that the operation planning can help RL in Appx. B.4.

4.5. Ablation Study

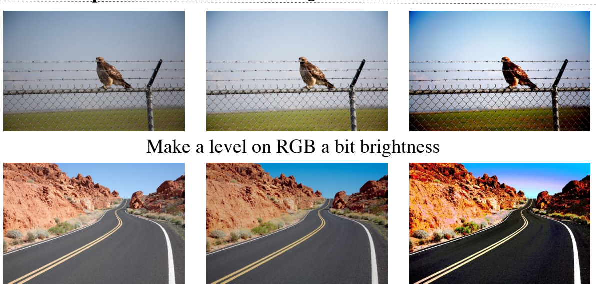
Due to space limit, the ablation study of different network structures is moved to Appx. A.3 and the investigation of alternative image loss is in Appx. A.1.

Trade-off between L1 and variance. We sample opera-

operation set	1	2	3	4	5	input
planning (train)	0.0521	0.0358	0.0198	0.0197	**0.0136**	0.1202
T2ONet (test)	0.1315	0.0857	0.0832	0.0853	**0.0770**	0.1190

Table 2. L1 distance to target image over different operation lists and operation orders on MIT-Adobe 5k dataset. Set 1 is planned over only brightness operation. Set 2 is planned over single parameter operations including brightness, contrast, saturation, sharpness. Set 3 is planned over the full operation list with the operation order fixed. Set 4 is planned over full operations with epsilon-greedy search. Set 5 is planned over the full operation list. Inputs represent the input image.

Input **Target** **Plan on Discriminator Result**



Make a level on RGB a bit brightness

Increase saturation of road, sky and mountains and make the sky more blue

Figure 7. Planning through a discriminator.

Input **Target** **Recovered Output**

Figure 8. Planning on local editing.

tion parameter α_t from $\mathcal{N}(\alpha_t; \mu_{\alpha_t}, \sigma_\alpha)$ while training the L1 loss. We set $\sigma_\alpha = Rh/3$, where R is the half range of the parameter, h is the gaussian width controller. Interestingly, the L1 and variance of T2ONet can be traded-off by adjusting σ_α. Fig. 5 manifests that the image variance can be enlarged by increasing h, but in turn, resulting in higher L1. The detailed result table is in Appx. A.2. Moreover, Fig. 6 shows that while all of the models are sensitive to requests, the model trained with larger h produces more diversified results.

Planning with different operation lists, operation orders and planning methods. According to both the planning and T2ONet editing performance in Tab. 2, set 1, 2, 5 shows that the performance substantially increases as the operation candidate list becomes larger. Planning with different single operation and different max step N is studied in Appx. A.5. Set 3 and 5 compare the difference between fixed and our searched operation order. It shows the searched order is slightly better than the fixed one for planning (might because the improvement space for planning is limited), but it will bring a larger improvement for T2ONet. Set 4 and 5 indicate that the original version is better than alternative ϵ-greedy policy [33], detailed in Appx. A.4.

4.6. Extensions of Planning Algorithm

Planning through a discriminator. We leverage a discriminator D that takes as input a pair of images and a request and outputs a score indicating the editing quality. Such D is pretrained with adversarial loss on T2ONet (see Appx. A.1 for detail). We define the new cost function as $\text{cost}(I) = 1 - D(I_0, I, Q)$, and apply it to Alg. 1. Interestingly, such planning can still produce some visually pleasing results, shown in Fig. 7. Although its quantitative results are worse than our default training performance, using a pretrained image-quality discriminator to edit an image brings a new perspective for image editing. Another advantage is its flexibility such that the same discriminator can be applied on a different set of operations while previous methods require retraining.

Planning for local edit. Our operation planning can generalize to local editing (*e.g*. "remove the man in the red shirt on the left"). Given the input and target image, we can use the pretrained panoptic segmentation network [35] to get a set of segments in the input image. With our planning algorithm (adding a new loop for segments, adding inpainting as one operation), we can get the pseudo ground truth, including the inpainting operation and its edited area, which can train a local editing network like [30]. Its full algorithm is described in the Appx. C.

5. Conclusion

We present an operation planning algorithm to reverse-engineer the editing through input image and target image, and can even generalize to local editing. A Text-to-Operation editing model supervised by the pseudo operation sequence is proposed to achieve a language-driven image editing task. We proved the equivalence of the image loss and the deterministic policy gradient. Comparison experiments manifest our method is superior to other GAN-based and RL counterparts on both MA5k-Req and GIER Images. The ablation study further investigates the trade-off between L1 and request-sensitivity and analyzes the factors that affect operation planning performance. Finally, we extend the operation planning to a discriminator-based planning and local edit.

Acknowledgments This work was supported in part by an Adobe research gift, and NSF 1813709, 1741472 and 1909912. The article solely reflects the opinions and conclusions of its authors but not the funding agents.

References

[1] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *CVPR*, 2016. 2

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 3

[3] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input/output image pairs. In *CVPR*, 2011. 5

[4] Yu Cheng, Zhe Gan, Yitong Li, Jingjing Liu, and Jianfeng Gao. Sequential attention gan for interactive image editing via dialogue. *arXiv preprint arXiv:1812.08352*, 2018. 2

[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 3

[6] Hao Dong, Simiao Yu, Chao Wu, and Yike Guo. Semantic image synthesis via adversarial learning. In *ICCV*, 2017. 2, 6

[7] Alaaeldin El-Nouby, Shikhar Sharma, Hannes Schulz, Devon Hjelm, Layla El Asri, Samira Ebrahimi Kahou, Yoshua Bengio, and Graham W Taylor. Tell, draw, and repeat: Generating and modifying images based on continual linguistic instruction. In *ICCV*, 2019. 2, 6

[8] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated planning and acting*. Cambridge University Press, 2016. 2, 4

[9] Rafael C Gonzales and Richard E Woods. Digital image processing, 2002. 3

[10] Ronghang Hu, Jacob Andreas, Trevor Darrell, and Kate Saenko. Explainable neural computation via stack neural module networks. In *ECCV*, 2018. 2

[11] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. In *ICCV*, 2017. 2

[12] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. In *CVPR*, 2017. 2

[13] Yuanming Hu, Hao He, Chenxi Xu, Baoyuan Wang, and Stephen Lin. Exposure: A white-box photo post-processing framework. *ACM Transactions on Graphics (TOG)*, 37(2):1–17, 2018. 2, 3, 5

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2, 6

[15] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Inferring and executing programs for visual reasoning. In *ICCV*, 2017. 2

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4, 6

[17] George Konidaris, Leslie Pack Kaelbling, and Tomas Lozano-Perez. From skills to symbols: Learning symbolic representations for abstract high-level planning. *Journal of Artificial Intelligence Research*, 61:215–289, 2018. 2

[18] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. Manigan: Text-guided image manipulation. In *CVPR*, 2020. 2, 6

[19] Ke Li, Tianhao Zhang, and Jitendra Malik. Diverse image synthesis from semantic layouts via conditional imle. In *ICCV*, 2019. 6

[20] Daqing Liu, Hanwang Zhang, Feng Wu, and Zheng-Jun Zha. Learning to assemble neural module tree networks for visual grounding. In *ICCV*, 2019. 2

[21] Ramesh Manuvinakurike, Jacqueline Brixey, Trung Bui, Walter Chang, Doo Soon Kim, Ron Artstein, and Kallirroi Georgila. Edit me: A corpus and a framework for understanding natural language image editing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018. 1

[22] Ramesh Manuvinakurike, Trung Bui, Walter Chang, and Kallirroi Georgila. Conversational image editing: Incremental intent identification in a new dialogue task. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 284–295, 2018. 1

[23] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision. In *ICLR*, 2019. 2

[24] Xiaofeng Mao, Yuefeng Chen, Yuhong Li, Tao Xiong, Yuan He, and Hui Xue. Bilinear representation for language-based image editing using conditional generative adversarial networks. In *ICASSP*, 2019. 2, 6

[25] Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. Pddl-the planning domain definition language, 1998. 2

[26] Seonghyeon Nam, Yunji Kim, and Seon Joo Kim. Text-adaptive generative adversarial networks: manipulating images with natural language. In *NuerIPS*, 2018. 2, 6

[27] John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965. 4, 6

[28] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 3

[29] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016. 2

[30] Jing Shi, Ning Xu, Trung Bui, Franck Dernoncourt, Zheng Wen, and Chenliang Xu. A benchmark and baseline for language-driven image editing. *arXiv preprint arXiv:2010.02330*, 2020. 1, 2, 5, 8

[31] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. 2014. 5

[32] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NeurIPS*, pages 3104–3112, 2014. 5

[33] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018. 8

[34] Hai Wang, Jason D Williams, and SingBing Kang. Learning to globally edit images with textual description. *arXiv preprint arXiv:1810.05786*, 2018. 1, 2, 5, 6

[35] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *CVPR*, pages 8818–8826, 2019. 8

[36] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In *NeurIPS*, 2018. 2

[37] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *CVPR*, 2018. 2