Learning by Planning: Language-Guided Global Image Editing

Appendix

Jing Shi
Ning Xu
Yihang Xu
Trung Bui
Franck Dernoncourt
Chenliang Xu

1University of Rochester
2Adobe Research

1{j.shi,chenliang.xu}@rochester.edu
1yxu74@u.rochester.edu
2{nxu,bui,dernonco}@adobe.com

Contents

A Ablation Study
  A.1. Different Image Loss
  A.2. Trade-off between L1 and Variance
  A.3. Effect of historical operations, images and attention for T2ONet
  A.4. Comparison of other possible planning methods
  A.5. Effect of different single operation lists and different maximum operation steps

B Reinforcement Learning
  B.1. Details of RL baseline
  B.2. Equivalence of image loss and DPG
  B.3. Algorithm for RL baseline
  B.4. Can operation planning benefit RL?

C Planning for Local Editing

D Time Analysis

E More advantages of T2ONet
  E.1. Resolution independent editing
  E.2. Inference with multiple possible output

F More visual results
  F.1. Comparison Methods
  F.2. T2ONet
  F.3. Operation Planning

G More Experiment Implementation Details

H Operation Implementation Details
  H.1. Brightness and Saturation
  H.2. Contrast
  H.3. Sharpness
  H.4. Tone and Color

I Languages for Image Variance Evaluation

J Dataset
  J.1. More Detail of MA5k-Req Collection Process
  J.2. Visualization of Dataset Samples

K User study details

A. Ablation Study

A.1. Different Image Loss

Inspired by the GAN-based image-to-image translation [12], we also try to apply adversarial loss \( L_{adv} \) using discriminator \( D \), whose structure is shown in Fig. 1. The adversarial loss is expressed as:

\[
L_{adv} = -\mathbb{E}_{(I_0,I_g)}[\log(D(I_0,I_g,Q))] - \mathbb{E}_{(I_0,I_T)}[\log(1 - D((I_0,I_T,Q)))].
\]  

(1)

Denote the whole parameter for the T2ONet as \( \Theta_G \) and discriminator as \( \Theta_D \), the objective for adversarial loss is \( \min_{\Theta_G} \max_{\Theta_D} L_{adv} \). The effect of L1 and adversarial loss is shown in Tab. 3. We observe that adding the image level loss can significantly improve T2ONet, because the operation supervision is trained in teacher forcing fashion, which easily accumulates error at each step. A supervision at the final image help correct the error at the final image. And without image supervision, the variance drops significantly, indicating the model has a very similar output for different requests. Moreover, the L1 loss is better than the adversarial loss. It might because adversarial loss is good at generating sharper and more detailed images [2, 5], but our operation will not reduce the detail/texture of the image, so
A.2. Trade-off between L1 and Variance

Table 2 shows the complete evaluation for the trade-off between L1 and variance.

A.3. Effect of historical operations, images and attention for T2ONet

Our standard T2O Cell takes in the previous operation and image. The comparison with only either of them is shown in Tab. 3, indicating that just image or operation performs no better than their combination. One exception is the variance for the only operation is better than combined, which means without the historical image as the feedback, the editing will be less controlled and be more diversified. Also, the attention mechanism help improve the performance according to Tab. 3.

A.4. Comparison of other possible planning method

Since our operation planning is based on greedy best-search, such greedy method does not guarantee an optimal solution. Inspired by the ε-greedy policy [10] applied in RL, we further compare a variant called ε-greedy operation planning to incorporate randomness to further approach optimal. The only difference is that there is 5% possibility that the operation is randomly selected, rather than the top choice.

A.5. Effect of different single operation lists and different maximum operation steps

We further study the comparison of only applying single operation. Table 4 presents the editing results of planning and T2ONet using only single operation. The most effective operations are “tone” and “color”, because they have 8 and 24 parameters, respectively, and thus have stronger editing ability than single-parameter operations. The results also suggest that the single-step editing results are worse than our proposed multi-step editing results. Also, we track the effect of different maximum operation steps, the planning results are shown in Tab. 5. From the perspective of planning, maximum steps 4, 5, 6 do not have much difference. This suggests that we could reduce the editing step or find the best trade-off between the editing effect and editing time complexity in the future.

B. Reinforcement Learning

B.1. Details of RL baseline

Now we reformulate the editing problem into a partial observed Markov decision process and introduce an RL baseline. Following the symbol notations and the problem formulation in the main paper that the editing process is a sequential action decision problem, we augmented with reward \( r_{t+1} \) for action \( a_t \), the problem can be reformulated into a Markov decision process and solved by RL. Following [4], the reward is set to indicate the incremental image quality, which is adapted as the reduction of the image cost

\[
    r_t = \text{cost}(I_{t-1}) - \text{cost}(I_t),
\]

where cost\((I)\) can be any image loss and is set as \(||I - I_g||\)\) in our experiment. Since the reward for the “END” action is hard to design (the reward in Eq. (2) is zero for “END” action), we set every episode fixed \( 5 \) steps \((T = 5) \) as [4]). The actions are sampled from policy \( \pi = (\pi_o, \pi_a) \), where \( \pi_o = P(o|s), \pi_a = P(a|o, s) \), leading to the trajectory \( \Pi = \{s_0, a_0, s_1, r_1, ..., s_T, r_T\} \). The \( P(o|s), P(a|o, s) \)
DPG using the following theorem:

\section*{Equivalence of image loss and DPG}

For the continuous policy \( \pi \), the goal is to optimize the objective \( J(\pi) = \mathbb{E}_{(I_0, Q) \sim P(D) \atop \alpha \sim \pi_\alpha} \sum_{t=0}^{T-1} G_{t+1} \nabla_{\theta_a} \log \pi_o(\alpha_t) \). \hfill (3)

For the continuous policy \( \pi_\alpha \), we resort to DPG \cite{2016arXiv160605978M}. Different from the common setting \cite{2016arXiv160605978M, 2017arXiv170307718P} 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 \( \alpha_t \), resulting in the DPG as

\[ \nabla_{\theta_a} J(\pi) = \mathbb{E}_{(I_0, Q) \sim P(D) \atop \alpha \sim \pi_\alpha} \sum_{t=0}^{T-1} G_{t+1} \nabla_{\theta_a} \alpha_t. \] \hfill (4)

In short, the major difference of our RL optimization from \cite{2017arXiv170307718P} is that we replace Q function approximated by neural network in \cite{2017arXiv170307718P} with \( G \) in both discrete and continuous policies, avoiding the complexity for training the Q network. The full algorithm for our RL baseline is shown in Alg. 1.

In our experiments, the sampling for \( o \) is based on \( \pi_\alpha \) with \( \epsilon \)-greedy policy where the \( \epsilon = 0.05 \). The sampling for \( \alpha \) is based on \( \pi_\alpha \) where the gaussian width controller \( h = 0.1 \). The other implementation details are the same with our main experiments.

\section*{Equivalence of image loss and DPG}

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

\section*{Theorem 1. The DPG for \( \alpha \) in Eq. (4) can be rewrite as}

\[ \nabla_{\theta_a} J(\pi) = \mathbb{E}_{(I_0, Q) \sim P(D) \atop \alpha \sim \pi_\alpha} \frac{\partial \cost(I_T)}{\partial \theta_a}. \] \hfill (5)

\section*{Proof.}

Substituting Eq. (2) and \( \gamma = 1 \), \( G_t \) can be simplified as

\[ G_t = \sum_{t=0}^{T-t} (\cost(I_t) - \cost(I_{t+1})) \]

\[ = \cost(I_1) - \cost(I_T). \] \hfill (6)

Since \( I_t \) is independent of \( \alpha_t \), we have

\[ \nabla_{\alpha_t} G_{t+1} = \frac{\partial \cost(I_t) - \cost(I_{t+1})}{\partial \alpha_t} = \frac{\partial \cost(I_T)}{\partial \alpha_t}. \] \hfill (7)

Therefore, the summation in Eq. (4) can be expressed as

\[ \sum_{t=0}^{T-1} \nabla_{\alpha_t} G_{t+1} \nabla_{\theta_a} \alpha_t = \sum_{t=0}^{T-1} \frac{\partial \cost(I_T)}{\partial \alpha_t} \frac{\partial \alpha_t}{\partial \theta_a} = \frac{\partial \cost(I_T)}{\partial \theta_a}. \] \hfill (8)

According to Eq. (8), Eq. (4) is equivalent to Eq. (5). \qed

\section*{Algorithm for RL baseline}

The full algorithm for our RL baseline is shown in Alg. 1.

\begin{algorithm}[!h]
\begin{algorithmic}[1]
\State \textbf{Input:} Training dataset \( D \); learning rate \( \beta \); max operation step \( N = 5 \)
\For {episode in 1 \colon M}
\State Sample \( I_0, Q, I_g \) from \( D \);
\State Sample one editing episode from \( \pi_o, \pi_o \); \{\( I_0, a_0, I_1, r_1, a_2, I_2, \ldots, I_r, r_T \} \};
\State \( \Delta_{\theta_o} J = \sum_{t=0}^{T-1} G_{t+1} \nabla_{\theta_o} \log \pi_o(a_t); \)
\State \( \theta_o \leftarrow \theta_o + \beta \Delta_{\theta_o} J; \)
\State \( \Delta_{\theta_o} J = -\frac{\partial \cost(I_T)}{\partial \theta_o}; \quad \# \cost(I) = ||I - I_g||_1 \)
\State \( \theta_o \leftarrow \theta_o + \beta \Delta_{\theta_o} J; \)
\EndFor
\State \( \theta_o \leftarrow \theta_o \)
\State \textbf{return} \( (\theta_o, \theta_o) \)
\end{algorithmic}
\end{algorithm}

\begin{table}[!h]
\centering
\begin{tabular}{lccccccc}
\hline
\textbf{Operation} & \textbf{Brightness} & \textbf{Contrast} & \textbf{Saturation} & \textbf{Sharpness} & \textbf{Tone} & \textbf{Color} & \textbf{Input} \\
\hline
planning (train) & 0.0521 & 0.0859 & 0.1037 & 0.1163 & \textbf{0.0277} & \textbf{0.0260} & 0.1202 \\
T2ONet (test) & 0.1315 & 0.1178 & 0.1163 & 0.1256 & \textbf{0.1006} & \textbf{0.1129} & 0.1190 \\
\hline
\end{tabular}
\caption{L1 distance to target image over different single operations on MA5k-Req dataset. Input represents the distance of the input image to the target image. Planning results are on the training set, and T2ONet results are on the testing set.}
\end{table}

\begin{table}[!h]
\centering
\begin{tabular}{lcccccccc}
\hline
\textbf{Max Step} & 1 & 2 & 3 & 4 & 5 & 6 & \textbf{Input} \\
\hline
Planning (train) & 0.0256 & 0.0145 & 0.0139 & 0.0137 & 0.0136 & 0.0136 & 0.1202 \\
\hline
\end{tabular}
\caption{L1 distance to the target image with different maximum editing steps on MA5k-Req dataset. Input represents the distance of the input image to the target image.}
\end{table}

is computed the same way as T2ONet. With the accumulated reward defined as \( G_t = \sum_{t=0}^{T-t} \gamma^t r_{t+1} \) (\( \gamma = 1 \) as \cite{2017arXiv170307718P}), the goal is to optimize the objective \( J(\pi) = \mathbb{E}_{(I_0, Q) \sim P(D) \atop \alpha \sim \pi_\alpha} \sum_{t=0}^{T-1} G_{t+1} \nabla_{\theta_a} \log \pi_o(\alpha_t) \).

According to Eq. (8), Eq. (4) is equivalent to Eq. (5). \qed
B.4. Can operation planning benefit RL?

Since the success of RL relies on the exploration of the action space, can the action sequence obtained from the operation planning algorithm help RL to better explore the action space, especially the continuous action? To answer this question, similar to [15], we firstly pretrain the model with the planned operations as supervision (same as T2ONet training loss), then finetune it using RL with only the target image supervision. The results in Tab. 6 show that the pretraining does not help RL much on MA5k-Req, but significantly benefit RL on GIER. As GIER has smaller size and more complex editing than FiveK, RL is struggling with the exploration of \( \alpha \). The pretrained model can initialize a good exploration and thus the RL can work on GIER.

C. Planning for Local Editing

Our operation planning can generalize to local editing. Given a zero-one image mask \( M \), we redesign the image editing function as \( I_{\text{out}} = o(I, \alpha) \odot M + I \odot (1 - M) \), where \( \odot \) is element-wise product; thus only the masked part is edited. Given \( K \) mask candidates, we can add an inner loop over all \( K \) mask candidates to further generate \( K \) edited images each time. In this case, the time complexity goes to \( O(NB|O|K) \). However, \( K \) can be removed if we know the grounded mask for each operation. Its full algorithm is described in Alg. 2. We use UPSNet [14] to obtain the mask candidates and use [6] for removing/inpainting operation. Given each operation with its region, it could also train our T2ONet augmented with the grounding model. Since this paper focuses on global operation planning, it will be left for future work.

D. Time Analysis

We compare the running time of T2ONet and Pix2pixAug [11] in Tab. 7. For T2ONet, the computing-intensive planning is a pre-processing step and only needs to be computed once, and our model shows faster train and test speed than Pix2pixAug, indicating that our method not only has better editing quality, but also is computational cheaper.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Pretrain</th>
<th>L1 ( \downarrow )</th>
<th>SSIM ( \uparrow )</th>
<th>FID ( \downarrow )</th>
<th>( \sigma \times 10^4 ) ( \uparrow )</th>
</tr>
</thead>
<tbody>
<tr>
<td>MA5k-Req</td>
<td>( \times )</td>
<td>0.1007</td>
<td>0.8283</td>
<td>7.4896</td>
<td>1.6175</td>
</tr>
<tr>
<td>MA5k-Req</td>
<td>( \checkmark )</td>
<td>0.0955</td>
<td>0.8330</td>
<td>7.1413</td>
<td>1.4672</td>
</tr>
<tr>
<td>GIER</td>
<td>( \times )</td>
<td>0.2286</td>
<td>0.3832</td>
<td>132.1785</td>
<td>0.3978</td>
</tr>
<tr>
<td>GIER</td>
<td>( \checkmark )</td>
<td>0.1052</td>
<td>0.8075</td>
<td>49.4183</td>
<td>1.0949</td>
</tr>
</tbody>
</table>

Table 6. The RL performance with and without operation-supervised pretrain on two datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>Planning (s)</th>
<th>Train (s)</th>
<th>Test (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pix2pixAug [11]</td>
<td>18.58</td>
<td>0.37</td>
<td>0.05</td>
</tr>
<tr>
<td>T2ONet</td>
<td>-</td>
<td>1.16</td>
<td>0.16</td>
</tr>
</tbody>
</table>

Table 7. The average running time comparison for GAN-based method Pix2pixAug [11] and our method. The training time is computed in batch size 64, and test and planning time are computed in batch size 1.

E. More advantages of T2ONet

E.1. Resolution independent editing

Our model will conduct resolution-independent editing and can produce the output with the same resolution as the input image. However, the GAN-based method suffers from generating high-resolution images, such comparison is shown in Fig. 3.

E.2. Inference with multiple possible output

We have discussed the trade-off between L1 and variance by sampling the operation parameter during training stage, and such variance is measured over the outputs edited from different requests, with the purpose of indicating the
T2ONet (\(-L1 + D\))

Request: Please increase the saturation.

Input

Target

Figure 2. Visualization for ablation study methods. T2ONet(-L1) is the modified version without L1 loss, T2ONet(-L1+D) is to replace the L1 loss to adversarial loss.

Figure 3. Compared with the GAN-based method GeNeVa and Pix2pixAug, although all the methods conduct the correct editing, our method has no pixel distortion and is independent to image resolution.

Figure 4. Visualization for diversified output given the same input and request by sampling the operation parameter at inference stage.

F. More visual results

F.1. Comparison Methods

Here we show the comparison visual result of Bilinear-GAN, TAGAN and ManiGAN in Fig 5. The visual results for ManiGAN is quite blur, and its L1, SSIM, FID are 0.1398, 0.5177, 157.4145 on MA5k-Req and 0.1834, 0.4938, 234.6784 on GIER, respectively. Therefore we did not do user study for this method.

F.2. T2ONet

More visual results for T2ONet on MA5k-Req and GIER are shown in Fig. 6 and Fig. 7, respectively.

F.3. Operation Planning

More visualization of the operation editing process is shown in Fig. 8.
decoder has hidden size 512. All the other FC layers output with a 512-d feature.

For operation planning, we adopt Nelder-Mead [7] for parameter optimization. And, for language-guided image editing, the training is alternatively in two losses. For odd iteration, we only optimize $L_o$ and $L_a$ in a teacher forcing fashion. For even iteration, we only optimize $L_{l1}$ using the previously generated action and image as the input for the next state. We take the top-1 operation with its parameters every step. The final image-level $L_{l1}$ can backward propagate gradients to the weights of T2OCell other than the weights of the FC layer for prediction of the operation $o$. Hence, in all ablation study of the T2ONet, we always need the loss of $L_o$ to supervise the operation selection. The model is trained on a single GPU with a 64 batch size.

**H. Operation Implementation Details**

We adopt six operations: brightness, saturation, contrast, sharpness, tone, and color. The operation modular network is composed of these operations in a fixed order if they are needed. With the input image $I$, parameter $p$, and output image $I'$, the implementation of operation submodules are illustrated as follows.

**H.1. Brightness and Saturation**

The hue, saturation, value in the HSV space of image $I$ are denoted as $H(I)$, $S(I)$, $V(I)$. Here $p$ is an unbounded scalar. Let $V'(I) = \text{clip}(1 + p \cdot V(I), 0, 1)$ and $S'(I) = \text{clip}((1 + p) \cdot S(I), 0, 1)$, the output image for brightness operation is

$$I' = \text{HSVtoRGB}(H(I), S(I), V'(I)),$$

and the output image for saturation operation is

$$I' = \text{HSVtoRGB}(H(I), S'(I), V(I)).$$

The HSVtoRGB is a differentiable function mapping the RGB space to HSV space, implemented via Kornia [8], and $\text{clip}(x, 0, 1)$ is a clip function to clip $x$ within 0 to 1.

**H.2. Contrast**

Contrast operation is controlled by a scalar parameter $p$, implemented following [4]. First compute the luminance of image $I$ as

$$\text{Lum}(I) = 0.27I_r + 0.67I_g + 0.06I_b,$$

where $I_r$, $I_g$, $I_b$ are the RGB channels of $I$. The enhanced luminance is

$$\text{EnhancedLum}(I) = \frac{1}{2} (1 - \cos(\pi \cdot \text{Lum}(I))),$$

and the image with enhanced contrast is

$$\text{EnhancedC}(I) = I \cdot \frac{\text{EnhancedLum}(I)}{\text{Lum}(I)}.$$

The output image $I'$ is the combination of the enhanced contrast and original image

$$I' = (1 - p) \cdot I + p \cdot \text{EnhancedC}(I).$$

**H.3. Sharpness**

The sharpness operation is implemented by adding to the image with its second-order spatial gradient [1], expressed as

$$I' = I + p \Delta^2 I,$$
where $p$ is a scalar parameter and $(\Delta^2 \cdot \cdot)$ is the Laplace operator over the spatial domain of the image. The Laplace operator is applied to each channel of the image.

**H.4. Tone and Color**

The tone and color operation follows curve representation [4]. The curve is estimated as piece-wise linear functions with $N$ pieces. The parameter $p = \{p_i\}_{i=0}^{M-1}$ is a vector of length $M$. With the input pixel $x \in [0, 1]$, the output
Figure 7. The visual results for T2ONet on GEIR dataset.
where \( Z = \sum_{i=0}^{N-1} p_i \). For tone operation, \( N = M = 8 \), the same \( f \) will apply to each of the RGB channels of the image \( I \). For color operation, three different \( f \) are applied individually to each of RGB channels. Each \( f(x) \) has \( N = 8 \), leading to \( M = 3N = 24 \).

### I. Languages for Image Variance Evaluation

The 10 different requests are as follows:

1. Decrease the brightness.
2. Increase the brightness.
3. Enhance the color.
4. Decrease the color.
5. Improve contrast.
6. Reduce contrast.
7. Increase saturation.
8. Reduce saturation.
9. Increase the brightness a little.
10. Increase the brightness a lot.

### J. Dataset

#### J.1. More Detail of MA5k-Req Collection Process

In this section we show the worker the input and target images, and let workers write the editing request. We deploy the annotation collection interface on Amazon Mechanical Turk involving totally 268 workers for FiveK. Each request annotation is $0.03, and we have the approvals for crowdsourcing.

We show the worker the input and target images, and let workers write the editing request. We deploy the annotation collection interface on Amazon Mechanical Turk involving totally 268 workers for FiveK and 197 workers for web images. Each request annotation is $0.03.

For quality control, we initially collect the language requests for a subset of image pairs, and manually select good workers depending on the annotation quality. Then we only allow good workers to annotate the full dataset.

#### J.2. Visualization of Dataset Samples

Some samples draw from MA5k-Req and GIER is shown in Fig. 9.
K. User study details

The interface of the user study is shown in Fig. 10.

References
