MaLP: Manipulation Localization Using a Proactive Scheme
– Supplementary material –

Vishal Asnani¹, Xi Yin², Tal Hassner², Xiaoming Liu¹
¹Michigan State University, ²Meta AI
¹{asnani_v, liuxm}@msu.edu, ²{yinxi, thassner}@meta.com

1. Implementation Details

Experimental Setup and Hyperparameters. We train MaLP for 150,000 iterations with a batch size of 4. For all of the networks, we use Adam optimizer except for the transformer which uses AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay $0.5e^{-5}$ and eps $1e^{-8}$. The learning rate is $1e^{-5}$ for all networks. The constraint weights are set as: $\lambda_1 = 100, \lambda_2 = 5, \lambda_3 = 4, \lambda_4 = 25, \lambda_5 = 25, \lambda_6 = 25, \lambda_7 = 50, \lambda_8 = 15, \lambda_9 = 20, \lambda_{10} = 50$. We use a template set size of 1 and template strength as 30% unless mentioned. All experiments are conducted on one NVIDIA K80 GPU.

Network Architecture. We show the network architecture of various components of MaLP in Fig. 1. The shared network consists of 1 stem convolutional layer and 4 convolution blocks. Each convolution block consists of convolutional and batch normalization layers followed by ReLU activation. The output of the shared network is given to $E_E$ and $E_C$, both having the same architecture with 3 convolution blocks and 1 stem convolutional layer. We use the transformer $E_T$ in the second branch of the framework where the ViT [6] architecture is adopted. The transformer consists of 6 encoder blocks, and a dropout of 0.1 is used. The features of the transformer are reshaped to the shape of the fakeness map $i.e.$ $1 \times 128 \times 128$. Finally, we use a classifier $C$ on the predicted fakeness maps to perform real vs. fake binary classification. The classifier has 8 convolution blocks, 1 stem convolutional layer, and 3 fully connected layers. We apply the ReLU activation between the layers.

GMs and dataset license information. We use a variety of face and generic GMs to show the effectiveness of MaLP. The information for all the GMs along with their training datasets, is shown in Tab. 1. For many GMs used by [1], we use the test images released by [1]. for the remaining GMs, we would release the test images for fair comparison of generalization benchmark by the future works. We also show more visualization samples of the predicted fakeness maps by MaLP in Fig. 2-5. All the fakeness maps are shown in "pink" cmap for better representation. We also indicate the cosine similarity between the predicted and ground truth fakeness maps. We observe that the fakeness maps for encrypted images have minimal bright regions. However, for fake images, MaLP is able to localize the modified regions well, considering the modified attributes/GMs are unseen in training.

The face datasets include CelebA [16] and CelebA-HQ [12], both of which don’t have any associated Institutional Review Board (IRB) approval. The authors for both datasets mention the availability of the dataset for non-commercial research purposes, which we strictly adhere to. For generic images datasets, we use Facades [25], COCO [2], Horse2Zebra [30], Summer2Winter [30], GTA2CITY [23], Edges2Shoes [11], Paris street-view [18], and Sketch-Photo [26] datasets. All the mentioned generic image datasets can be used for non-commercial research purposes, as mentioned by the authors, and we use the datasets for the same purposes.

Image Editing Degradations. We apply several image editing degradations to the test set to verify the robustness of MaLP. The details of these operations are listed below:

1. JPEG compression: We compress the image with the compression quality of 50%.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>GMs</th>
</tr>
</thead>
<tbody>
<tr>
<td>CelebA [16]</td>
<td>STGAN [14], AttGAN [8], StarGAN [4], GANanimation [22], CouncilGAN [18], ESRGAN [27], GDWCT [3]</td>
</tr>
<tr>
<td>Horse2Zebra [30]</td>
<td>AutoGAN [29]</td>
</tr>
<tr>
<td>GTA2CITY [23]</td>
<td>UNIT [15]</td>
</tr>
<tr>
<td>Paris Street-view [18]</td>
<td>Cont_Enc [20]</td>
</tr>
<tr>
<td>Sketch-Photo [26]</td>
<td>DualGAN [28]</td>
</tr>
</tbody>
</table>

Table 1. List of GMs along with their training datasets
Figure 1. Network architecture for different components of MaLP. (a) Shared network, (b) Encoder \(E_E\) and CNN network \(E_C\), (c) Classifier \(C\), (d) Transformer \(E_T\), and (e) Transformer encoder block.

### Table 2. Ablation for localization loss.

<table>
<thead>
<tr>
<th>Loss</th>
<th>CS ↑</th>
<th>PSNR ↑</th>
<th>SSIM ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS</td>
<td>0.9356</td>
<td>22.16</td>
<td>0.7114</td>
</tr>
<tr>
<td>CS + (L_1)</td>
<td>0.9230</td>
<td>18.98</td>
<td>0.6614</td>
</tr>
<tr>
<td>CS + SSIM + (L_2)</td>
<td>0.9211</td>
<td>19.12</td>
<td>0.6816</td>
</tr>
<tr>
<td>CS + SSIM + (L_1)</td>
<td>0.8777</td>
<td>14.04</td>
<td>0.3712</td>
</tr>
<tr>
<td>CS + SSIM</td>
<td>0.9394</td>
<td>23.020</td>
<td>0.7312</td>
</tr>
</tbody>
</table>

2. Blur: We apply the Gaussian blur with a filter size of \(7 \times 7\).

3. Noise: We apply a Gaussian noise with zero mean and unit variance.

4. Low-resolution: We resize the image to half the original resolution and restore it back to the original resolution using linear interpolation.

### Potential Societal Impact

The problem of manipulation localization is crucial from the perspective of media forensics. Localizing the fake regions not only helps in the detection of these fake media but, in the future, can also help recover the original image that the GM has manipulated. We also show that MaLP can be used as a discriminator to improve the quality of GMs. While this is an interesting application of MaLP, it can be a possibility that the GMs become more robust to our framework, decreasing the localization performance if the training of the GM is done from scratch.

### 2. Additional Experiments

#### Localization Loss
We show the importance of manipulation loss (defined in Eq. 8) in Sec. 4.6. We perform an ablation to formulate the loss of fakeness maps for manipulated images. As shown in Tab. 2, we try experimenting with various loss functions i.e. cosine similarity (CS), \(L_1\), \(L_2\) and structural similarity index measure (SSIM). Using just the CS loss results in better performance compared to combining it with \(L_1\) or \(L_2\) loss. We observe a huge deterioration in performance when using \(L_1\) loss. This can be explained as PSNR and SSIM are directly related to mean squared error which is optimized by either an \(L_2\) or SSIM loss. Finally, adopting an SSIM loss with CS loss results in a better performance as both of them are more related to the metrics, making it easier for MaLP to converge.

#### Comparison with Baseline
Due to the limited GPU memory, we conduct proactive training with one GM only because the GM needs to be loaded to the memory and used on the fly. On the other hand, passive methods can be trained on multiple GMs because the image generation
processes are conducted offline. As shown in Tab. 3, [10] trains on images manipulated by 7 different GMs, unlike MaLP, which is trained on images manipulated by only 1 GM. We show the performance on three GMs, which are seen for [10], but unseen for MaLP. MaLP performs better even though these GMs’ images are not seen in training. Therefore, even though the training of MaLP is limited by 1 GM, it can achieve better generalization to other GMs proving the effectiveness of proactive schemes.

Multiple Attribute Modifications. Instead of training on bald attribute modification by STGAN, we train and test MaLP on multiple attribute modifications. These include bald, bangs, black hair, eyeglasses, mustache, and smile manipulation. We show the results in Tab. 4. MaLP performs better for all the attribute modifications compared to the passive method [10]. We also observe an increase in cosine similarity compared to when MaLP is trained on only bald attribute modification. This is expected, as the more types of modifications MaLP sees in training, the better it learns to localize.

Transformer Architecture Ablation. We ablate various parameters of the transformer to select the best architecture for manipulation localization. We experiment with parameters that include optimizer, depth i.e. number of blocks, and dropout. We only use the transformer branch and switch off the CNN branch during training. The results are shown in Tab. 5. We observe that the localization performance is almost the same when using the transformer to predict fakeness maps. However, the detection accuracy has a significant impact. Having dropout does increase the performance for detection and localization. Further, using the weighted Adam optimizer is more beneficial than using the vanilla Adam optimizer. Therefore, we adopt the architecture of the transformer with 6 blocks and optimize it with a weighted Adam optimizer. Finally, we also include the dropout to achieve the best performance for localization and detection.

Table 5. Ablation study for transformer architecture.

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>Depth</th>
<th>Dropout</th>
<th>Cosine similarity†</th>
<th>Accuracy†</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adam</td>
<td>6</td>
<td>0.1</td>
<td>0.8839</td>
<td>0.9514</td>
</tr>
<tr>
<td>AdamW</td>
<td>1</td>
<td>0.0</td>
<td>0.8825</td>
<td>0.9647</td>
</tr>
<tr>
<td>AdamW</td>
<td>1</td>
<td>0.0</td>
<td>0.8826</td>
<td>0.9680</td>
</tr>
<tr>
<td>AdamW</td>
<td>3</td>
<td>0.0</td>
<td>0.8830</td>
<td>0.9705</td>
</tr>
<tr>
<td>AdamW</td>
<td>6</td>
<td>0.1</td>
<td><strong>0.8848</strong></td>
<td><strong>0.9856</strong></td>
</tr>
</tbody>
</table>

References

[18] Ori Nizan and Ayellet Tal. Breaking the cycle - colleagues are all you need. In CVPR, 2020. 1
Figure 2. Visualization of fakeness maps for different attribute modifications by STGAN. (a) Real image, (b) encrypted image, (c) manipulated image, (d) ground-truth $M_{GT}$, (e) predicted fakeness map for encrypted images, and (f) predicted fakeness map for manipulated images. We also show the cosine similarity between the predicted and ground-truth fakeness map below (f). All face images come from SiWM-v2 data [7].

Figure 3. Visualization of fakeness maps for different attribute modifications by STGAN. (a) Real image, (b) encrypted image, (c) manipulated image, (d) ground-truth $M_{GT}$, (e) predicted fakeness map for encrypted images, and (f) predicted fakeness map for manipulated images. We also show the cosine similarity between the predicted and ground-truth fakeness map below (f). All face images come from SiWM-v2 data [7].


[7] SiWM-v2 data
Figure 4. Visualization of fakeness maps for manipulation by DRIT. (a) Real image, (b) encrypted image, (c) manipulated image, (d) ground-truth $M_{GT}$, (e) predicted fakeness map for encrypted images, and (f) predicted fakeness map for manipulated images. We also show the cosine similarity between the predicted and ground-truth fakeness map below (f).
Figure 5. Visualization of fakeness maps for manipulation by GauGAN. (a) Real image, (b) encrypted image, (c) manipulated image, (d) ground-truth $M_{GT}$, (e) predicted fakeness map for encrypted images, and (f) predicted fakeness map for manipulated images. We also show the cosine similarity between the predicted and ground-truth fakeness map below (f).