Synthesizing Image with High-Quality Segmentation Mask by Prompting Large Vision Model

Supplementary Material

In this supplementary document we include:

- Pseudocode 1 the GenPrompt algorithm described in Section 2.2 of the main paper.
- Additional implementation details.
- Ablation study and additional qualitative results.

5. Implementation details

Training procedure

To evaluate the effectiveness of the synthetic dataset, we assess the performance of existing segmentation models trained on it. As in [12, 14] we ensure a fair comparison by applying the same training procedure used for real data. Formally, given a segmentation model $\mathcal{F}$, the predicted segmentation mask Y for the generated image I is obtained as:

$$Y = \mathcal{F}(I). \tag{11}$$

The loss $\mathcal{L}$ used to update parameters in model $\mathcal{F}$ is calculated via the cross-entropy loss function $\mathcal{L}_{\text{CE}}$:

$$\mathcal{L} = \mathcal{L}_{\text{CE}}(Y, I_{\text{mask}}). \tag{12}$$

Hyperparameter setting

We set the threshold of certainty α to 0.8 for extracting candidate prompts. The number of iteration N is set to 100 to find the optimal point prompt set from the candidate set . For each object of interest in the image, we select 3 point prompts $(K = 3)$. The other parameters used to generate the attention map are kept consistent with those in [12]. We utilized the text prompt provided in [12] to generate synthetic images using Stable Diffusion version 2.1-base [13]. We evaluate the model's performance on the validation sets of the VOC and COCO datasets. The training procedure is based on the MMsegmentation framework [5]. We use the AdamW optimizer with a learning rate of $1e-4$, weight decay of $1e-4$, and train for $20,000$ iterations with a batch size of 8. We follow the standard MMSegmentation framework for the other model hyperparameters. We conducted the training using a server with four Tesla V100 GPUs, each with 32GB of memory, an Intel Xeon E5-2698 processor, and 256GB of RAM. For synthetic image and segmentation mask generation, we ran the process in parallel over eight V100 GPUs, which took ten hours to generate $40,000$ data samples.

6. Additional Results

6.1. Ablation study

For all experiments in the ablation study, we use the DeepLabv3 model and evaluate on the VOC dataset, with the same configuration as described in Section 3.1.

Table 2 shows the impact of varying point prompt quantity on the final performance. As has been discussed in Section 2.2, using a small number of point prompts lead to the performance degradation. However, we observed that increasing the number of points beyond a certain threshold did not yield notable performance improvements. This aligns with the findings of [6] regarding the impact of point prompt numbers on segmentation performance. We selected three point prompts to achieve a balance between performance and computational cost.

Table 3 showcases the results of employing various point prompt generation techniques. As detailed in Section 2.2 and Figure 3, straightforward strategies exhibit limitations in producing high precise segmentation mask for the whole object. On the other hand, our proposed GenPrompt method, which generates point prompts with maximized diversity, achieves the highest quality segmentation masks from the given probability maps.

We explore the impact of generated image quantity on segmentation model performance in Table 4. We observed there was a positive but diminishing marginal effect of increasing synthetic data quantity on performance. This suggests that the approach may have reached a limitation, where additional data provides minimal benefit. This also aligns with the findings of [12] regarding the effect of synthetic data size.

The effect of threshold α to extract candidate points is present in Table 5. Note that, unlike baseline methods that apply the threshold directly to binarize the attention map, we use the threshold only to extract the candidate point prompts. The purpose of this is to filter out points with low certainty (usually the point at the object boundary), to ensure the the selected point prompts belonging to the target object. We observe that the performance is not much sensitive to the choice of α. However, if choose the threshold too high, candidate prompts will not cover most part of object and will tend to concentrate on local area. The value of alpha as 0.8 achieve the best performance of 65.1 mIOU.

Algorithm 1 Generating Point Prompts with Maximum Diversity.

Input:
- Number of object classes M
- Probability map $P = \{p_{ij}^m\}$, where each p_{ij}^m represents the certainty of pixel at location (i, j) belonging to object class m

Output:
- Point prompts set for M object classes $\mathcal{S} = \{\mathcal{S}^m\}_{m=1}^M$

Initialize $\mathcal{S}$ as an empty set for all classes $m \in 1, ..., M$
for $m = 1$ *to* M **do**
  foreach *point* (i, j) **do**
    if $m = \arg\max_\eta(p_{ij}^\eta)$ **then** $\hat{p}_{ij}^m \leftarrow p_{ij}^m$ `// Obtain coarse probability map for class m;`
    else $\hat{p}_{ij}^m, \leftarrow 0$;
  end
  Define $\mathcal{C}^m$ as set $\{(i, j) \mid \hat{p}_{ij}^m > 0\}$ and initialize D_{max} as 0 `// Construct the candidate set`
  for $n = 1$ *to* N **do**
    $\mathcal{S}_{\text{tmp}} \leftarrow$ Randomly select K points that have a certainty above α from $\mathcal{C}^m$
    $D_{\text{tmp}} \leftarrow$ Compute harmonic mean of distances between all pairs of points in $\mathcal{S}_{\text{tmp}}$ `// Refer to Equation 10`
    if $D_{tmp} > D_{max}$ **then** Update D_{max} with D_{tmp} and assign $\mathcal{S}_{\text{tmp}}$ to $\mathcal{S}_{\text{sel}}$;
  end
  Assign set $\mathcal{S}_{\text{sel}}$ to $\mathcal{S}^m$
end
return $\mathcal{S}$

# point prompts	mIoU (%)
2	52.3
3	65.1
4	64.8

Table 2. Impact of different number of point prompts.

Prompts selection	mIoU (%)
Single point	38.8
Multiple points	43.6
Random	61.0
Maximum diversity	65.1

Table 3. Performance of various point prompts generation methods.

# images	mIoU (%)
20k	64.0
30k	65.0
40k	65.1

Table 4. Impact of different number of generated images.

α	mIoU (%)
0.7	64.8
0.8	65.1
0.9	64.6

Table 5. Analysis of α.

6.2. Additional qualitative results

We present additional qualitative results in Figures 5, 6 and 7. These examples showcase the superiority of our proposed prompting technique over the thresholding baseline.

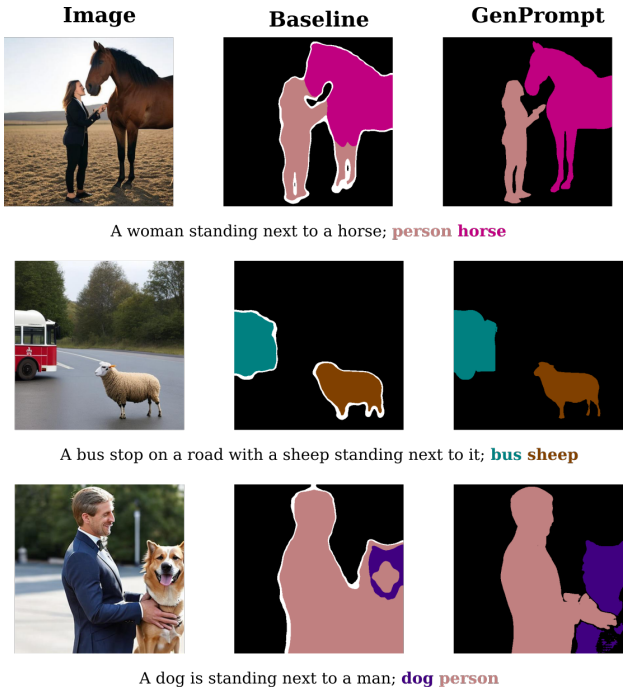

Figure 5. **Additional qualitative results, refer Sec. 3.3 for details.**

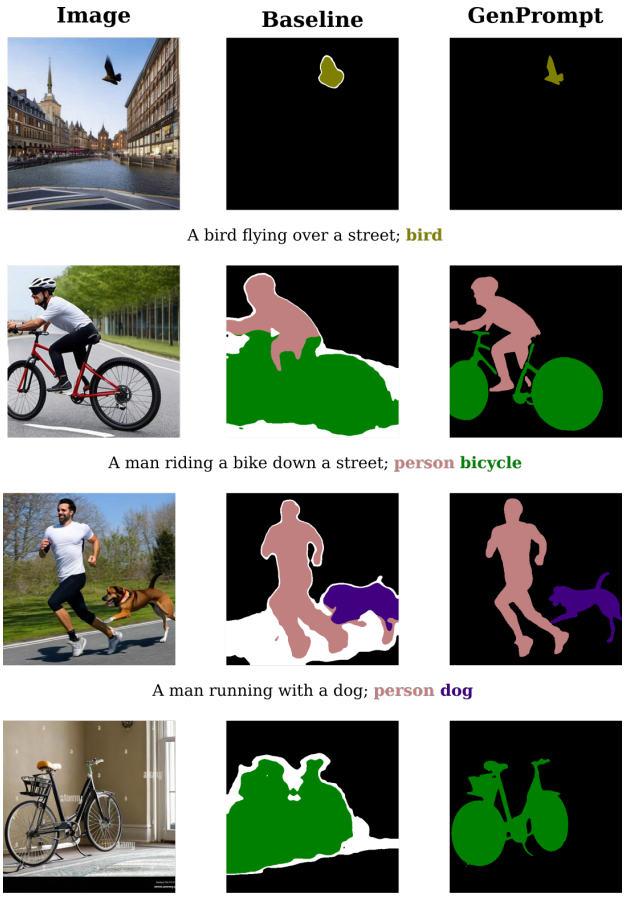

Figure 6. **Additional qualitative results, refer Sec. 3.3 for details.**

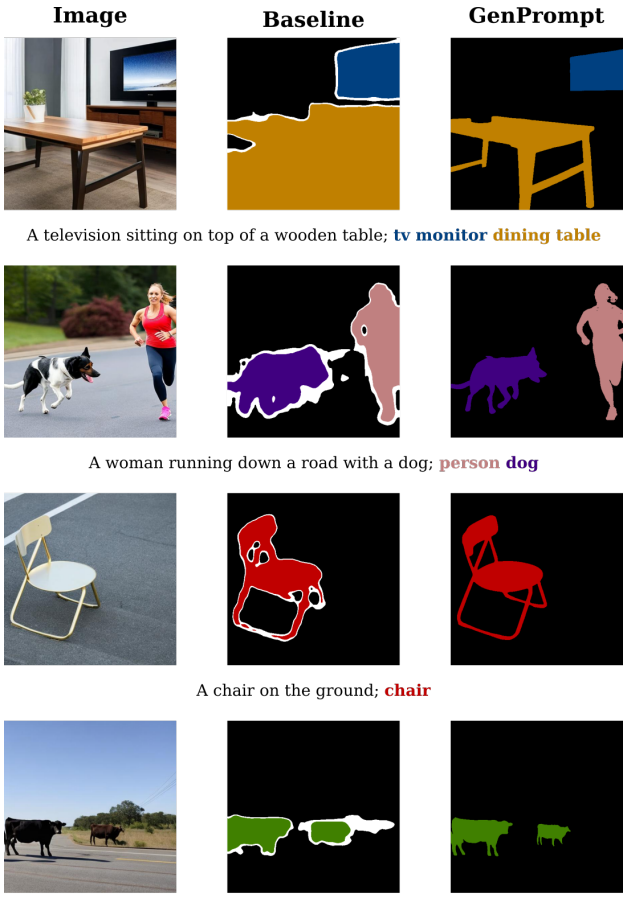

Figure 7. **Additional qualitative results, refer Sec. 3.3 for details.**