Unifying Automatic and Interactive Matting with Pretrained ViTs

Supplementary Material

6. Appendix

We provide the following content in this supplementary:

- Pseudo code for our SMat.
- More visualization results of our method.
- Complete quantitative results of the comparison.

A. Pseudo Code

In this section, we provide the pseudo-code for our method, shown in Algorithm 1.

Algorithm 1: Training Process of SMat

```
def train(images, prompts, alpha_gt):
  """images: [b,3,h,w] prompts: [b,1,h,w]"""
  img_feat, cls_token = image_encoder(images)
  """img_feat: [b,c,h/s,w/s]"""
  """cls_token: [b,c,1,1]"""
  state = not_all_zeros(prompts)
  """state: [b,1] (1-inter 0-auto)"""
  prompt_feat = downsample(prompts) * img_feat
  # generate candidate feature
  F_can = state * prompt_feat + (1 - state) *
      cls_token
  """F_can: [b,c,1,1]"""
  # update candidate feature
  F_can = cross_attn(F_can, img_feat)
  sim_map = sim_func(F_can, img_feat)
  # predict and backward
  alpha_pred = sim_decoder(images, img_feat,
      sim_map)
  loss = objective_func(alpha_pred, alpha_gt)
  return loss
```

Note that all user prompts are converted into masks of the same resolution as images, where all regions covered by user prompts are highlighted with a value of 1. The process is illustrated in Fig. 8.

Then we obtain the state conditioned on whether a user prompt is given for an image. If with an empty prompt, the state will be set to 0, as shown in the bottom part in Fig. 8. If the prompt is not all zeros, the state will be set to 1 standing for the interactive mode. The candidate feature is generated based on the state, adaptively changing between the prompt feature and the class token.

For the update of the candidate feature, we employ a 8-head cross-attention layer for the interaction.

The objective function used in our method is an l_1 loss modified from that in ViTMatte [33], and a laplacian loss. In ViTMatte, the l_1 loss supervises unknown and known regions separately as

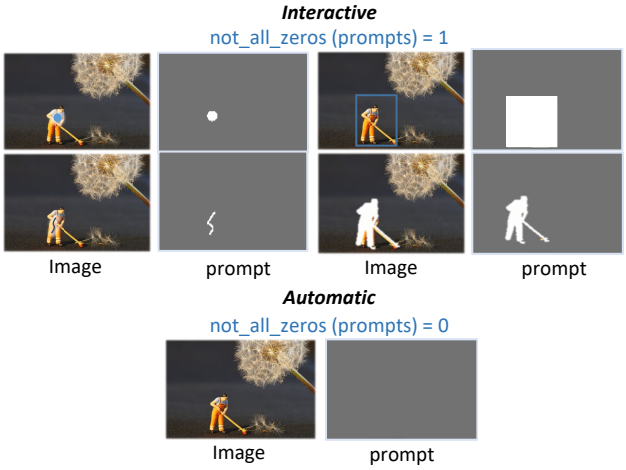

Figure 8. **Transform user prompt into a mask form.**

$$\mathcal{L}_{\text{separate } l_1} = \frac{1}{|\mathcal{U}|}\sum_{i\in\mathcal{U}}|\hat{\alpha}_i - \alpha_i| + \frac{1}{|\mathcal{K}|}\sum_{i\in\mathcal{K}}|\hat{\alpha}_i - \alpha_i| \ , \quad (3)$$

while we separately supervise foreground and background regions as

$$\mathcal{L}_{\text{separate } l_1} = \frac{1}{|\mathcal{F}|}\sum_{i\in\mathcal{F}}|\hat{\alpha}_i - \alpha_i| + \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}|\hat{\alpha}_i - \alpha_i| \ . \quad (4)$$

It is because ViTMatte has a trimap as its additional input, therefore it only needs to address the unknown region, while our setting requires extracting the foreground without the trimap.

B. Similarity Map Visualization

As shown in Fig. 9, we visualize the similarity map for each image generated by an empty user prompt and point user prompt, respectively. With an empty prompt, the class token will act as candidate feature, which represents saliency information within an image. Therefore, by comparing similarity with image features, the salient instances are highlighted. With point prompt, the prompt feature will work as candidate feature, so that it can locate the target object based on the average feature of the prompt area. It is obvious that even in the conjunction of two instances, our method can have the ability for distinction, which shows its superior instance awareness. As shown in Fig. 10, our probabilistic similarity map has a positive effect on the final prediction of details, while the binary map can mislead the prediction and have a negative effect.

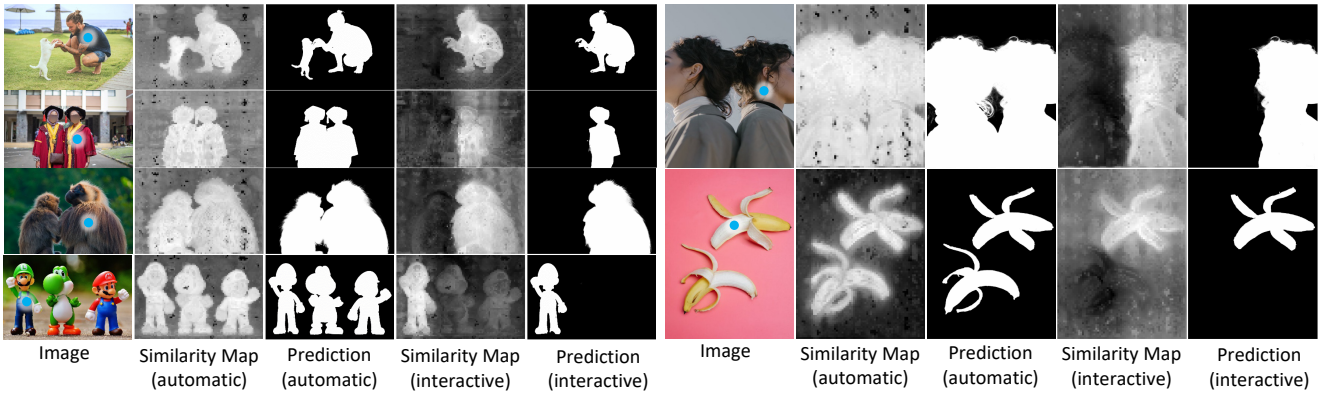

Figure 9. **Visualization of similarity map and alpha matte in different scenarios.**

automatic methods	category	AM2K (animal)					P3M-500-NP (human)					AIM-500 (natural)				
		SAD	MSE	MAD	Grad	Conn	SAD	MSE	MAD	Grad	Conn	SAD	MSE	MAD	Grad	Conn
GFM	animal	11.11	0.0031	0.0065	9.26	6.94	111.98	0.0613	0.0649	32.37	16.26	95.84	0.0505	0.0573	44.78	11.62
PPM	human	23.06	0.0096	0.0131	15.61	8.14	13.38	0.0042	0.0078	13.05	9.36	97.36	0.0512	0.0581	50.10	12.05
PPM-ViTAE	human	37.84	0.0189	0.0221	18.59	6.98	7.80	0.0017	0.0045	9.61	5.38	109.69	0.0584	0.0651	50.71	6.39
AIM	natural	32.03	0.0124	0.0186	29.52	13.80	65.57	0.0404	0.0512	82.72	42.24	48.09	0.0183	0.0285	47.58	21.74
SMat-auto	natural	16.84	0.0047	0.0098	17.98	10.08	18.93	0.0064	0.0110	19.07	10.83	34.30	0.0129	0.0203	31.49	13.98

Table 8. **Complete quantitative comparison with different automatic matting methods.** The results are generated with official models provided by the authors.

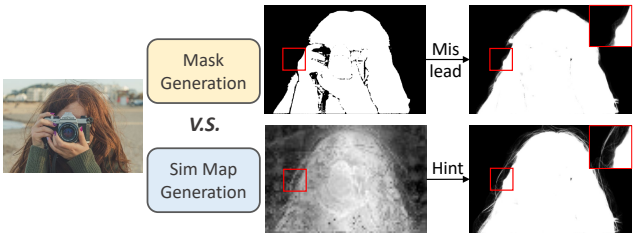

Figure 10. **Effect of different guidance type.**

Method	Support Visual Prompt	AIM-500	
		SAD	MSE
TIMINet(ICCV'21)	trimap	29.18	0.0092
SIM(CVPR'21)	trimap	27.07	0.0088
MatteFormer(CVPR'22)	trimap	26.87	0.0087
ViTMatte(arXiv'23)	trimap	17.21	0.0038
MGM(CVPR'21)	mask	71.91	0.0268
MG-Wild(CVPR'23)	mask	16.72	0.0030
MatAny(arXiv'23)	point,scribble,bbox,mask	124.36	0.0639
MatAny†(manually correction)		27.83	0.0093
SMat(Ours)	none,point,scribble,bbox,**mask**	22.53	0.0070

Table 9. Comparison with prompt-specific methods.

C. More Quantitative Results

Comparison with Prompt-specific Methods.

As shown in Table **??**, our method achieves comparable performance with the prompt-specific methods. Here we use mask as the prompt for our method to align with the mask-level guidance, different from the bbox used in the manuscript. Note that as a unified, prompt-agnostic method, this comparison maybe unfair to our method. However, the results further demonstrate the effectiveness and potential of our method.

Pretraining strategies and model volumes. To figure out the effect of the pretraining stage, we train SMat with different pretraining strategies and different variants of the pretrained model. As depicted in Table 10, DINOv2 demonstrates an obvious improvement compared with DINO, which can be attributed to its strong semantics and generalization ability. Also, the larger variant will lift the performance to a new height, however, it will lead to higher complexity and computation. Therefore, to balance efficacy and effectiveness, we choose DINOv2-S as the pretraining model for our SMat.

Generalization on class-specific automatic matting. Here we report the complete form of the Table 2 in the main text, the results are illustrated in Table 8. Category-specific methods can only address their target domain, while our

	Params (M)	AIM-500 prompt	SAD	MSE	AIM-500 prompt	SAD	MSE
dino-S [1]	26.5	none	37.96	0.0140	box	40.84	0.0155
dino-B [1]	96.6	none	36.55	0.0131	box	36.46	0.0140
dinov2-S [22]	26.9	none	34.30	0.0129	box	26.63	0.0083
dinov2-B [22]	97.4	none	30.62	0.0106	box	26.48	0.0081

Table 10. **Effect of pretrain strategies and model volume.** To seek the balance between efficacy and effectiveness, we select dinov2-S in our model.

sim_func	RefMatte-RW100 SAD	MSE	MAD
bmm	50.57	0.0260	0.0290
cos_sim	44.18	0.0223	0.0252

Table 11. **Effect of different similarity functions.**

method can provide stable performance across all foreground types.

Ablation study on similarity function. Here we conduct an ablation study on similarity function during the similarity comparison between candidate feature and image features, the results are shown in Table 11. One can see that the cosine similarity function is more suitable for this process, we attribute this to its implicit normalization contained in the computation so that the decoder receives the relative similarity instead of absolute similarity.

D. More Qualitative Results

Here we demonstrate more results of our method, the predictions are illustrated in Fig. 11 and Fig. 12.

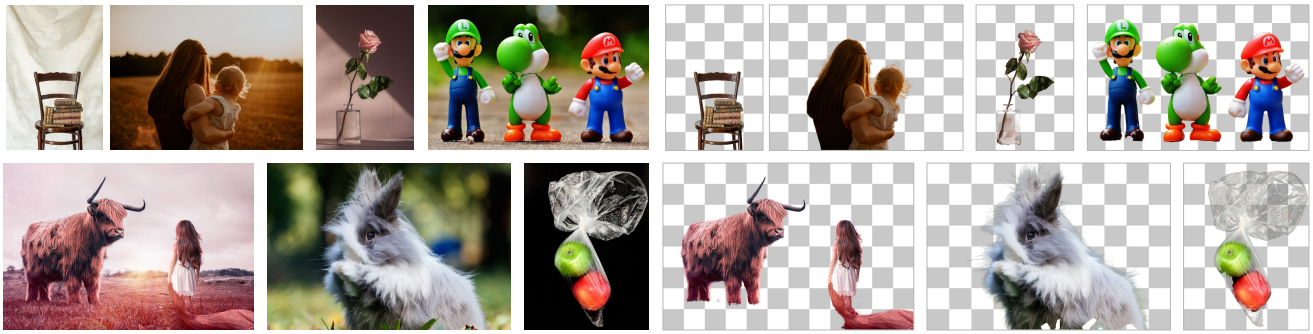

Figure 11. **More qualitative results of automatic prediction of our method.**

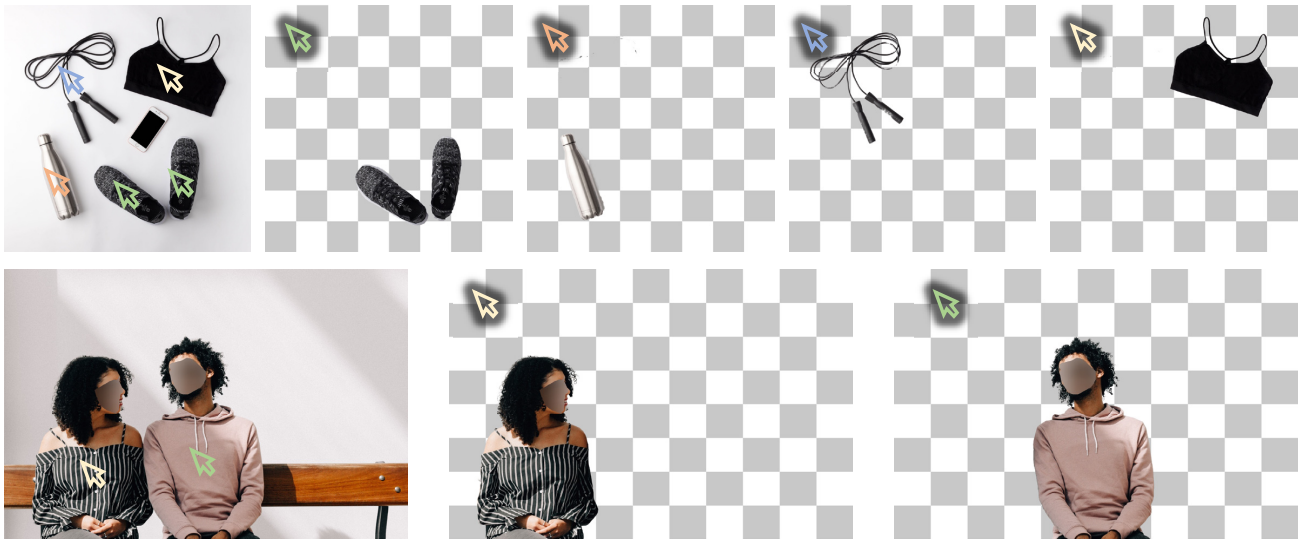

Figure 12. **More qualitative results of interactive prediction of our method.**