HiMTok: Learning Hierarchical Mask Tokens for Image Segmentation with Large Multimodal Model

Supplementary Material

A. Limitations

While HiMTok enables large multimodal models to acquire native (referring) image segmentation capabilities in a concise and natural manner, the current work has some limitations. (1) The length of predicted mask tokens is predefined. LMMs are not able to determine it adaptively according to object shape complexities. (2) The current model is relatively passive for object segmentation, due to the use of passive segmentation training data. We need to specify referring expressions to clarify the expected objects, rather than let the model itself segment all objects of interest at once. (3) It appears challenging for fine-grained region segmentation in the current version (See Appendix F). The lack of multi-scale feature design may cause the loss of fine-grained features.

B. Multi-grained mask labels

The multi-grained mask labels are important in the hierarchical mask loss. We have tried to use only the final mask label to supervise each granularity level, but the loss is relatively high due to the insufficient representation by few mask tokens. This also causes unstable training and even side influence on the general capability of LMM. In experiments, we find that few mask tokens are usually de-tokenized into Gaussian distribution maps, which inspire us to make multi-grained mask labels by Gaussian blurring.

Under the setting of mask token length to 32, we first choose a full-level (*i.e.*, 32 tokens) sequence and 3 random levels that are sampled by $p_l = \frac{1}{l+8}, 1 \leq l < 32$. For each partial level l, we produce a kernel following the 2D Gaussian distribution function $\mathcal{N}(\mu, \sigma)$, where $\mu = 0$, and $\sigma = \frac{100}{l+1} - 2$. Then the kernel is applied to the full-level mask image to obtain the mask label at level l. Fig. 8 visualizes some examples with different granularity levels.

C. Finetuning SAM

Despite the fact that LMM equipped with HiMTok, without relying on segmentation foundation models, has achieved state-of-the-art performance on various segmentation tasks, further improvements can still be expected by feeding the mask from our de-tokenizer into SAM [26]. This post-refinement module can be defined as a mapping $\mathcal{R} : (\mathcal{I}, \mathcal{M}_{in}) \rightarrow \mathcal{M}_{out}$, where $\mathcal{M}_{in}$ is the output mask by our HiMTok-equipped LMM, and $\mathcal{I}$ is the input image. However, we find that the native SAM given $\mathcal{M}_{in}$ tends to segment fine-grained parts of objects or generate mask maps

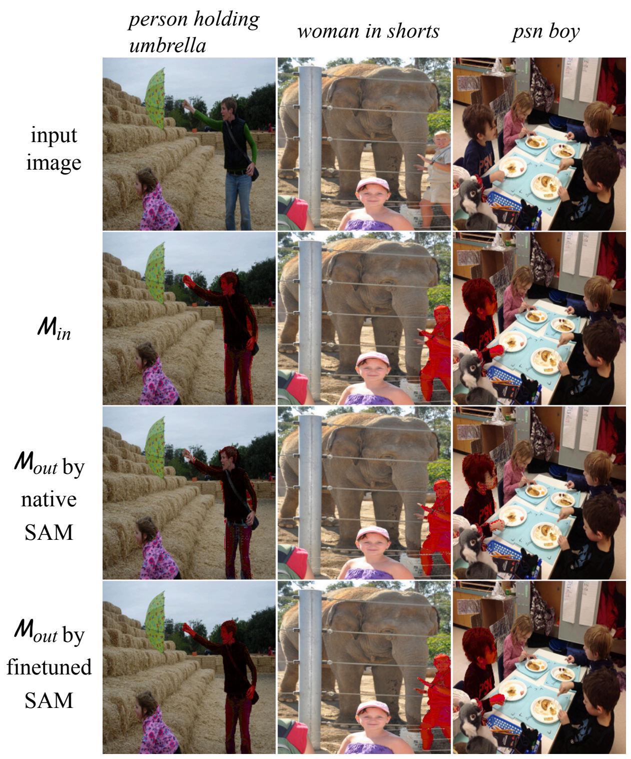


Figure 7. Further improvements by finetuned SAM.

with holes, even when the given mask clearly covers large and unambiguous regions, as visualized in Fig. 7.

To adapt to our case, we finetune the mask decoder and the mask convolution layers in SAM. The input mask $\mathcal{M}_{in}$ is augmented in two aspects. On the one hand, the number of mask tokens passed to the de-tokenizer is randomly reduced so that the fine-grained shape details may be lost. On the other hand, the de-tokenized mask is processed by random morphological augmentation, including dilation and erosion. As a result, the finetuned SAM is able to refine imperfect masks. Qualitative results are shown in Fig. 7. The finetuned SAM improve the edge details of the segmentation masks.

D. More details on training

In stage-3, some datasets are down-sampled. Details are listed in Tab. 9. The initial learning rate is $4e-5$ in stage 2, $2e-5$ in stage 3, and $8e-6$ in task finetuning. The GPU hours (Nvidia A800) for our 3 stages are: 192, 1920 and 640.

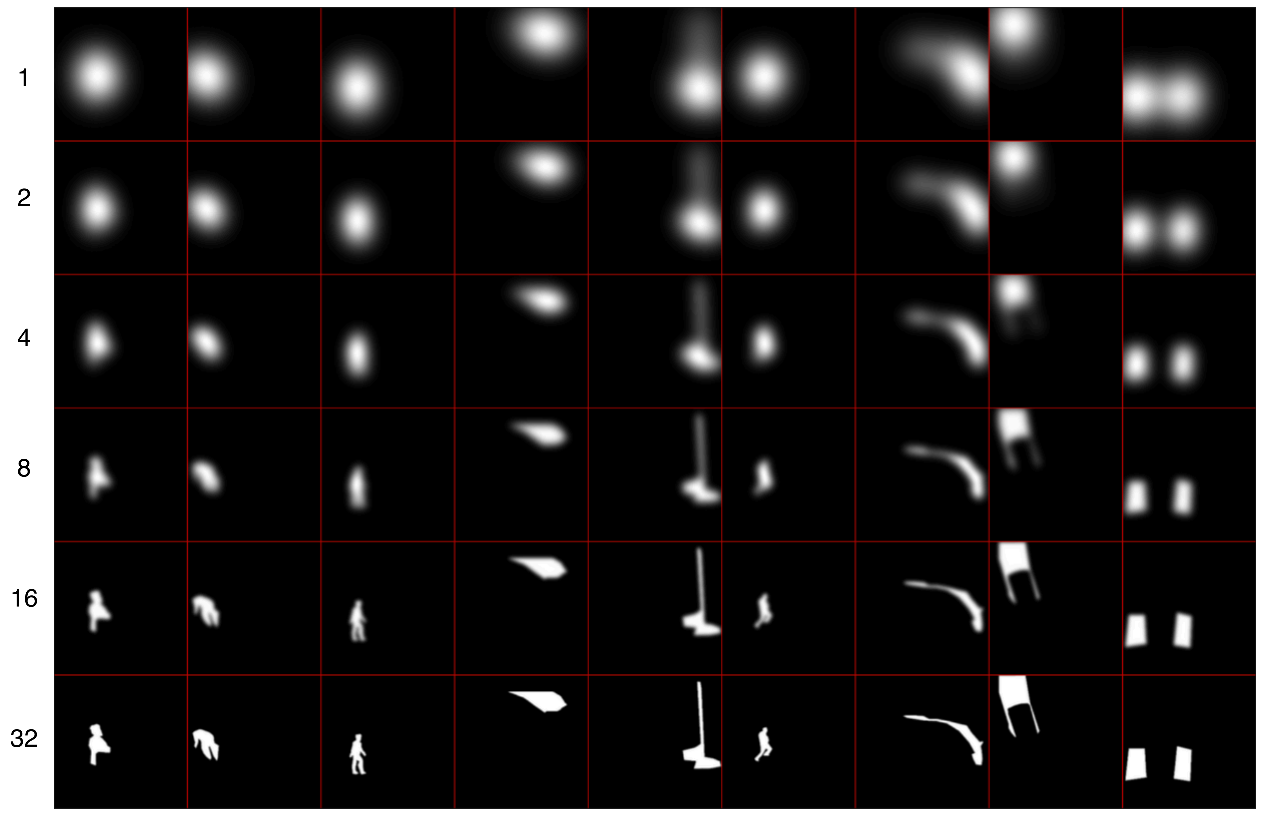


Figure 8. The Gaussian-blurred mask label at different levels.

Table 9. The amount of samples in down-sampled datasets for stage-3.

Dataset	stage-2	stage-3
SA1B	1M	250K
COCO-Rem	350K	35K
COCO-Stuff	500K	50K
COCO-Panoptic	233K	116K
PartImageNet	20K	4K
Objects365	450K	90K

E. Mask Perception

One key feature of our method is the consistence of mask representation in LLM input and output. The image segmentation tasks are all text-to-mask. As a supplement, we devise a mask-to-text task, Mask Perception[†], which aims for fine-grained understanding. Mask Perception (MaP) mirrors RES: models are required to choose a matched expression, given an image, an object mask and several expression options.

[†]https://huggingface.co/datasets/yayafengzi/Mask_Perception

The training and test sets are built based on RefCOCO/+/g [40, 72]. We randomly sample some examples and reverse the positions of masks and expressions in multiple-choice questions. One or more positive options are possible in the training set, while only one positive option is available in the test set to simplify evaluation. Negative expression options are selected from other objects in the same image or from different images. This approach encourages the model to distinguish the distinctive features of various parts. Statistics are shown in Tab. 10.

We have tried to compare our method with PSALM [78] which performs interactive segmentation well. However, we found that PSALM does not follow instructions well in our MaP test set. Therefore, we compared our method both with and without the MaP training set. As shown in Tab. 11, our method inherently has a good perception of input masks. With the addition of the MaP training set, the mask perception capability is significantly improved.

F. Results on fine-grained regions

Given that HiMTok decodes masks directly via mask tokens, a natural concern is whether it can maintain high seg-

Table 10. Statistics of Mask Perception. The data source is RefCOCO/+/g.

data split	source	single/multiple choices	No.
Training	train	multiple	190k
Test	val & test	single	10k

Table 11. Accuracy on mask perception.

w/ MaP	×	✓
Acc	63.1	81.8

mentation quality for fine-grained regions without leveraging the fine-grained features of the original image.

Here we show the results on small object segmentation in RefCOCO/+/g. Objects whose mask areas occupy less than 4% of the image are considered as small, resulting in 12.8% samples. Shown in Tab. 12, the cIoUs of ours still have significant priority compared to PSALM [78]. However, the cIoU scores fall significantly behind the overall performance (Tab. 2), which highlights a common challenge.

We also review the segmentation boundaries. Bfscore (a boundary-aware F1 metric) on RefCOCO (val) is reported: our method gets 0.927, which is competitive to PSALM (0.936). PSALM integrates Mask2Former that is favored by multi-scale features. We believe there are rooms for future exploration in our paradigm.

G. Results on general image understanding

Here, we list additional results on general image understanding. Our model is not finetuned on these tasks.

We compare methods on MME [19] Perception, VQAv2 [20], and POPE [32]. As shown in Tab. 13, our method is comparable to state-of-the-art LMMs, except in the areas of landmarks, artwork, and OCR, where we do not specially utilize corresponding training data in this work. We can conclude that $\text{LMM}_{\text{HiMTok}}$is a comprehensional and general LMM.

H. More visualizations

Fig. 9 shows more interesting and challenging cases. Fig. 10 illustrates how we implement referring image segmentation in conversations. This is what we do when evaluating our model on gRefCOCO, ReasonSeg, and open-vocabulary segmentation.

I. Prompt design

We prepared plentiful prompt templates for instruction tuning on segmentation and visual grounding.

For the bidirectional information flow between segmentation and grounding, Tabs. 14 and 15 list templates for

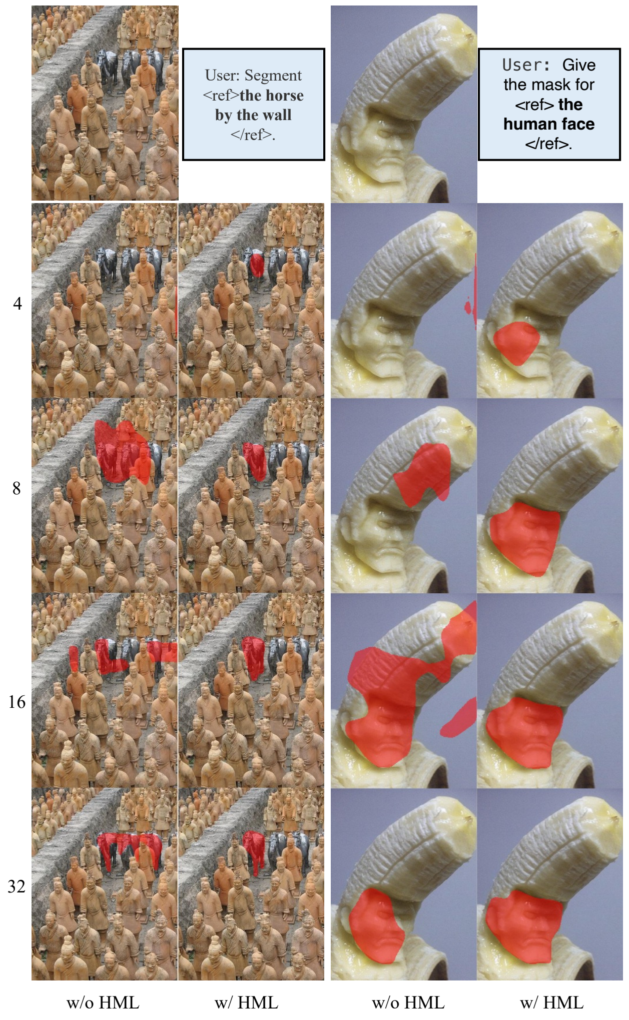

User: Segment <ref>the horse by the wall </ref>.
User: Give the mask for <ref> the human face </ref>.
4
8
16
32
w/o HML
w/ HML
w/o HML
w/ HML

Figure 9. More examples on the coarse-to-fine mask token representation with and without HML.

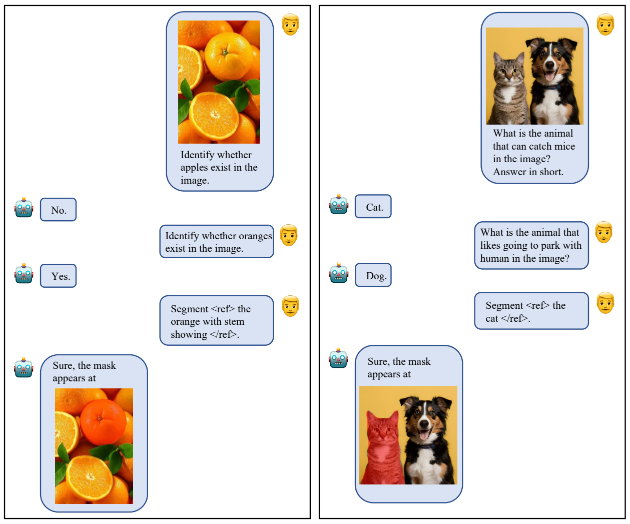
mask-to-box, and Tabs. 16 and 17 are for coordinate-to-mask. Tab. 18 list templates for segmentation-only responses. Templates in Tab. 19 are used to specify the mask token length from LMM. If visual grounding is the only target without mask tokens, we can refer to Tab. 20.

Table 12. Results on small object segmentation.

	Ref COCO			Ref COCO+			Ref COCOg	
	val	testA	testB	val	testA	testB	val	test
PSALM [78]	64.50	68.43	57.31	45.58	55.89	38.14	51.10	48.78
ours	67.15	75.00	60.34	56.81	63.06	48.29	57.71	55.99

Table 13. Results on general image understanding.

Method	MME										VQAv2	POPE
	existence	count	position	color	posters	celebrity	scene	landmark	artwork	OCR		
PSALM [78]	-	-	-	-	-	-	-	-	-	-	62.3	80.3
InternVL2.5-8B [11]	**200**	**170**	163	**180**	**169**	**140**	155	**172**	**160**	**178**	-	**90.6**
LMM$_{\text{HiMTok}}$-8B	**200**	160	**166**	**180**	164	132	**163**	132	120	88	**75.9**	86.8


Identify whether apples exist in the image.
No.
Identify whether oranges exist in the image.
Yes.
Segment <ref> the orange with stem showing </ref>.
Sure, the mask appears at

What is the animal that can catch mice in the image? Answer in short.
Cat.
What is the animal that likes going to park with human in the image?
Dog.
Segment <ref> the cat </ref>.
Sure, the mask appears at

Figure 10. Referring image segmentation in conversation.

Table 14. Templates of instruction for segmentation then grounding.

- "First create the segmentation mask for <ref>{}</ref>, then identify its bounding box."
- "Begin by generating the mask for <ref>{}</ref>, followed by determining its box coordinates."
- "Start with the detailed mask of <ref>{}</ref>, then locate its containing box."
- "Initially segment <ref>{}</ref>, then find its bounding box region."
- "First produce the mask for <ref>{}</ref>, then specify its box location."
- "Commence with the mask of <ref>{}</ref>, then determine its box boundaries."
- "Start by segmenting <ref>{}</ref>, then outline its bounding region."
- "First extract the mask of <ref>{}</ref>, then mark its box coordinates."
- "Begin with the precise mask for <ref>{}</ref>, then identify its box location."
- "Initially create the mask for <ref>{}</ref>, then define its bounding box."

Table 15. Templates of response for segmentation then grounding.

- "Certainly, you can find the mask at {}, and the box is represented as <box>{}</box>."
- "Of course, the mask located is {}, while the box is shown as <box>{}</box>."
- "Absolutely, the mask is situated at {}, with the box described as <box>{}</box>."
- "Sure, the mask appears at {}, and here's the box: <box>{}</box>."
- "Indeed, the mask can be found at {}, with the box marked as <box>{}</box>."
- "OK, the mask is at {}, and the box is indicated as <box>{}</box>."
- "Affirmative, you have the mask at {}, and the box is designated <box>{}</box>."
- "Got it, the mask is at {}, and you will see the box like <box>{}</box>."
- "Sure, the mask appears at {}, and here is the box represented: <box>{}</box>."
- "Yes indeed, find the mask at {}, and the box outlined as <box>{}</box>."

Table 16. Templates of instruction for box/point-prompted segmentation in SA1B.

- "Generate the segmentation mask for the object located within the bounding box <box>{}</box>and at the points {} in the provided image."
- "Extract the mask for the object inside the <box>{}</box>bounding box and corresponding to the points {} in the given image."
- "Create the mask for the object found in the region defined by <box>{}</box>and identified by the points {} in this image."
- "Identify the object mask within the bounding box <box>{}</box>and marked by the points {} in the image."
- "Acquire the segmentation mask of the object enclosed by <box>{}</box>and situated at the points {} in the image."
- "Determine the mask for the object that is located within the area specified by <box>{}</box>and identified by the points {} in the image."
- "Produce the mask for the object situated in the <box>{}</box>box and at the points {} within the image."
- "Locate the mask for the object that lies inside the <box>{}</box>boundaries and corresponds to the points {} in the image."
- "Segment the mask for the object found within the bounds of <box>{}</box>and marked at the points {} in the given image."
- "Outline the mask of the object residing within the box <box>{}</box>and corresponding to the points {} in the image."

Table 17. Templates of instruction for point-prompted segmentation in SA1B.

- "Generate the segmentation mask for the object located at the points {} in the provided image."
- "Extract the mask for the object that corresponds to the points {} in the given image."
- "Create the mask for the object identified by the points {} in this image."
- "Identify the object mask corresponding to the points {} in the image."
- "Acquire the segmentation mask of the object marked at the points {} in the image."
- "Determine the mask for the object located at the specified points {} in the image."
- "Produce the mask for the object situated at the points {} within the image."
- "Locate the mask for the object that is identified by the points {} in the image."
- "Segment the mask for the object found at the points {} in the given image."
- "Outline the mask of the object located at the specified points {} in the image."

Table 18. Templates of response for segmentation only.

- "Certainly, the mask is located at {}."
- "Of course, you can find the mask at {}."
- "Absolutely, the mask is available at {}."
- "Sure, the mask is positioned at {}."
- "Indeed, the mask appears at {}."
- "OK, the mask is situated at {}."
- "Affirmative, the mask can be found at {}."
- "Got it, the mask is at {}."
- "Sure, the mask is present at {}."
- "Yes indeed, you will find the mask at {}."

Table 19. Templates of instruction for segmentation with specified token lengths.

- "First create the segmentation mask for <ref>{}{}</ref>{} by {} tokens, then identify its bounding box."
- "Begin by generating the mask for <ref>{}{}</ref>{} by {} tokens, followed by determining its box coordinates."
- "Start with the detailed mask of <ref>{}{}</ref>{} by {} tokens, then locate its containing box."
- "Initially segment <ref>{}{}</ref>{} by {} tokens, then find its bounding box region."
- "First produce the mask for <ref>{}{}</ref>{} by {} tokens, then specify its box location."
- "Commence with the mask of <ref>{}{}</ref>{} by {} tokens, then determine its box boundaries."
- "Start by segmenting <ref>{}{}</ref>{} by {} tokens, then outline its bounding region."
- "First extract the mask of <ref>{}{}</ref>{} by {} tokens, then mark its box coordinates."
- "Begin with the precise mask for <ref>{}{}</ref>{} by {} tokens, then identify its box location."
- "Initially create the mask for <ref>{}{}</ref>{} by {} tokens, then define its bounding box."

Table 20. Templates of for visual grounding without mask tokens.

Instructions

- "Please provide the bounding box of <ref>{}</ref>."
- "Locate the bounding box for <ref>{}</ref>."
- "Identify the box coordinates of <ref>{}</ref>."
- "Determine the bounding region for <ref>{}</ref>."
- "Find the box boundaries of <ref>{}</ref>."
- "Mark the box location of <ref>{}</ref>."
- "Specify the box coordinates for <ref>{}</ref>."
- "Outline the bounding box of <ref>{}</ref>."
- "Can you show the box for <ref>{}</ref>?"

Responses

- "Certainly, the box is located at <box>{}</box>."
- "Of course, here's the bounding box: <box>{}</box>."
- "The requested box coordinates are <box>{}</box>."
- "Found the bounding region: <box>{}</box>."
- "Here's the box location: <box>{}</box>."
- "The object's box is marked as <box>{}</box>."
- "Bounding box determined: <box>{}</box>."
- "Located the boundaries at <box>{}</box>."