Supplemental Material for MAOAM: Unified Object & Material Selection with Vision-Language Models

JADEN PARK, University of Wisconsin-Madison, USA
VALENTIN DESCHAINTRE, Adobe Research, United Kingdom
JASON KUEN, Adobe Research, USA
KANGNING LIU, Adobe Research, USA
ILIYAN GEORGIEV, Adobe Research, United Kingdom
KRISHNA KUMAR SINGH, Adobe Research, USA
YONG JAE LEE, Adobe Research, USA
MICHAEL FISCHER, Adobe Research, United Kingdom

ACM Reference Format:
Jaden Park, Valentin Deschaintre, Jason Kuen, Kangning Liu, Iliyan Georgiev, Krishna Kumar Singh, Yong Jae Lee, and Michael Fischer. 2026. Supplemental Material for MAOAM: Unified Object & Material Selection with Vision-Language Models. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '26), July 19–23, 2026, Los Angeles, CA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3799902.3811186

In this supplementary material, we provide additional details on the datasets used to train our method as well as implementation and model details. We also provide more qualitative evaluation results and ablation studies that have been deferred due to limited space.

1 Dataset Details

We provide the number of source images and annotations for both train and validation splits for all datasets we use.

1.1 Material Datasets

The material selection datasets provide click- and text-based prompts for material selection with precise material masks.

RealMat consists of 7,848 images and 395 images in its training and validation sets, resulting in 46,646 and 2,214 material annotations for train and validation splits, respectively.

SynMat consists of 5,532 images and 352 images, which are frames sampled from videos, in its training and validation sets, which results in 54,315 and 3,071 material annotations for train and validation splits, respectively.

SAMa consists of 1,292 images and 141 images, which are also video frames, as its train and validation data. This results in 3,294 and 346 material annotations, for train and validation splits, respectively.

As a whole, our material dataset consists of ~104K and ~5.6K annotations for train and validation splits. Fig. 1 provides more visual examples of our material dataset.

1.2 Object and Entity Datasets

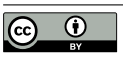
RefCOCO. We use the RefCOCO, RefCOCO+, and RefCOCOg datasets for text-based referring object selection. RefCOCO provides short, conversational referring expressions with relative spatial terms (e.g., "left of"). RefCOCO+ forbids location-based expressions, requiring appearance-based descriptions instead. RefCOCOg provides fewer but richer descriptions per object, with higher linguistic complexity. These datasets provide diverse object referring expressions that complement our material descriptions. RefCOCO contains 16,994 images, RefCOCO+ contains 16,992 images, and RefCOCOg contains 21,899 images. In total, the RefCOCO family provides approximately 56K training, 4.3K validation, and 5.6K test annotations. We use the official train, validation, and test splits.

EntitySeg. The EntitySeg [Qi et al. 2023] dataset provides referring prompts for click-based object selection. It consists of ~37K high-quality object selection masks from ~8K real-world images collected from datasets such as COCO [Lin et al. 2014], ADE20K [Zhou et al. 2017] and LAION-400M [Schuhmann et al. 2021].

The original dataset contains small masks that are difficult to reliably annotate with star overlays. We filter out invalid masks and masks smaller than 0.3% of the image area. After filtering, the dataset consists of 7,887 training images and 263 validation images, resulting in ~37K training and ~2.7K validation annotations.

Combined, our material and object selection training data consists of ~197K masks with varying selection criteria, material descriptions, and various orientations due to datasets that have been sampled from video frames.

Authors' Contact Information: Jaden Park, University of Wisconsin-Madison, USA, jadenpark@cs.wisc.edu; Valentin Deschaintre, Adobe Research, United Kingdom, deschain@adobe.com; Jason Kuen, Adobe Research, USA, kuen@adobe.com; Kangning Liu, Adobe Research, USA, kangningl@adobe.com; Iliyan Georgiev, Adobe Research, United Kingdom, igeorgiev@adobe.com; Krishna Kumar Singh, Adobe Research, USA, krishsin@adobe.com; Yong Jae Lee, Adobe Research, USA, yongl@adobe.com; Michael Fischer, Adobe Research, United Kingdom, mifischer@adobe.com.



SIGGRAPH Conference Papers '26, Los Angeles, CA, USA

ACM ISBN 979-8-4007-2554-8/2026/07
https://doi.org/10.1145/3799902.3811186

RealMat - hand-collected data | SynMat - synthetic, rendered data | SAMa

Fig. 1. Additional examples from our material datasets.

2 Training Details

We provide training details and hyperparameters for both backbone model configurations, as well as their architecture.

2.1 Architecture and Hyperparameters

We train MAOAM on two backbone configurations: Sa2VA (Qwen2.5-VL-7B [Bai et al. 2025] + SAM 2 [Ravi et al. 2024]) and GLaMM [Rasheed et al. 2024] (LLaVA-v1.5-7B [Liu et al. 2024] + SAM [Kirillov et al. 2023]). We list the hyperparameters below.

LLaVA-v1.5 and Qwen2.5-VL Architecture. LLaVA-1.5 [Liu et al. 2024] pairs a frozen CLIP ViT-L/14 image encoder with a Vicuna LLM [Chiang et al. 2023] via a two-layer MLP projector that maps visual features into the LLM's token embedding space; the model is trained on image-text instruction data with the projector and LLM trainable. Qwen2.5-VL [Bai et al. 2025] follows the same encoder–projector–LLM paradigm, with the addition of a native resolution ViT projected into the Qwen2.5 LLM backbone.

GLaMM Training and Inference. We train from the GLaMM-GranD-Pretrained checkpoint for 15 epochs. We use a linear learning rate decay schedule with minimum learning rate $1e-6$ and warm-up for the first 100 iterations. The initial learning rate is $2e-5$ for full VLM training and $3e-4$ for LoRA [Hu et al. 2022] (rank 8, alpha 16). We use AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.95$. For both models, we use the mask binary cross entropy loss and DICE loss for mask losses, and cross entropy loss for language modeling. We set $\lambda_{\text{BCE}} = \lambda_{\text{DICE}} = 1.5$ and $\lambda_{\text{CE}} = 0.5$ for GLaMM training.

For standard fine-tuning, we use a batch size of 4 and for LoRA fine-tuning, we use a batch size of 8. Since the VLM backbone is LLaVA, we train both the MLP adapter and the LLM, for both standard and LoRA fine-tuning cases. One epoch on our 190K Material and Object dataset takes approximately 8 hours on 8 A100 GPUs. During training, GLaMM-based MAOAM requires ~50GB VRAM for training and ~30GB VRAM during inference. Evaluating 1,000 images takes approximately one hour on 8 GPUs.

Sa2VA Training and Inference. We train from the Sa2VA-7B model checkpoint trained with Qwen2.5-VL-7B as the VLM backbone and SAM 2 as the selection head. Qwen2.5-VL-7B requires significantly more GPU VRAM compared to LLaVA-v1.5, and hence we train the model with a batch size of 1, and gradient accumulation steps of 4. Similar to GLaMM, we use AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.999$. We follow the default loss weights for Sa2VA, which are $\lambda_{\text{BCE}} = 2.0, \lambda_{\text{DICE}} = 0.5$. We use LoRA training with LoRA rank of 128 and alpha 256, while keeping only the MLP adapter trainable, which is the default fine-tuning setup for Qwen2.5-VL. One epoch training of Sa2VA model on our 190K Material and Object data takes approximately 12 hours on eight A100 GPUs. During training, Sa2VA-based MAOAM requires ~70GB VRAM for training and ~50GB VRAM during inference. Evaluating 700 images takes approximately one hour on 8 GPUs.

For all of our experiments, we train GLaMM for 15 epochs and Sa2VA for 10 epochs, resulting in a comparable wall-clock time of approximately 120 hours on eight A100 GPUs. Finally, we note that MAOAM's inference is slightly faster than the baseline models, since it does not require additional modules to encode the positional information (e.g., GLaMM's region encoder), which we pass via the star-overlay in our framework.

2.2 Detailed Task Formulation

Multi-task Training. Each data point in our material selection data consists of three tasks: click- and text-based selection, and VQA questions. Hence, we formulate the loss function as a multi-task loss, where the click-selection, text-selection, and VQA tasks are weighted $\lambda_{\text{click}} = 0.4, \lambda_{\text{ref}} = 0.4, \lambda_{\text{vqa}} = 0.2$. For single task case, i.e., RefCOCO or EntitySeg, where only $\mathcal{L}_{\text{ref}}$ and $\mathcal{L}_{\text{click}}$ are computed, respectively, we set the loss weights to 1.

Star overlay. During training, we randomly place 1-5 stars of size 32×32 pixels on the input image (1024×1024). We empirically observe that the model is sensitive to star locations, especially for thin selection areas. To ensure the star overlay provides a clear signal while also making the model robust to boundary cases, we erode the target area's binary mask using MaxPool2D with kernel size $r = 8$ to ensure most stars are included in the target area. When sampling multiple stars, we define boundary regions via erosion and place stars on boundary pixels with probability 0.5, and ensure that the stars are sufficiently far away from each other.

Finally, the color of the star overlay is dynamically determined to have the highest contrast from the region the star is being overlaid on to (for visualizations in the paper, we use a default white star for visibility). In case we place multiple stars, the first star's color is used throughout. There are a total of 10 star colors, and we use the same logic during model inference as well.

System prompt for material selection. We provide example prompts used for each task during training. All material-related prompts include a task prompt that clarifies the distinction between material and appearance variations. For each template, we generate about 3 to 7 paraphrased variants for diversity during training.

Table 1. Comprehensive evaluation on object-centric datasets. We report performance across RefCOCO, RefCOCO+, and RefCOCOg on their respective validation and test splits (text-based object selection), as well as EntitySeg (click-based object selection). MAOAM consistently shows competitive performance despite being trained jointly on material dataset.

	RefCOCO (Text)						**RefCOCO+ (Text)**						**RefCOCOg (Text)**				**EntitySeg (Click)**	
	Val		TestA		TestB		Val		TestA		TestB		Val		Test		Val	
Method	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑
SAM3	0.433	0.472	0.374	0.411	0.459	0.493	0.329	0.364	0.350	0.387	0.325	0.355	0.422	0.456	0.387	0.418	0.664	0.748
LISA	0.732	0.797	0.761	0.822	0.696	0.767	0.638	0.702	0.720	0.779	0.589	0.662	0.665	0.733	0.689	0.759	0.209	0.249
GLaMM	0.616	0.692	0.655	0.736	0.620	0.692	0.521	0.597	0.525	0.604	0.480	0.551	0.603	0.679	0.620	0.698	0.364	0.423
Sa2VA	0.781	0.840	0.819	0.874	0.765	0.826	0.729	0.782	0.787	0.842	0.671	0.731	0.749	0.810	0.758	0.819	0.435	0.495
MAOAM (Ours)	**0.809**	**0.895**	**0.835**	**0.910**	**0.770**	**0.870**	**0.744**	**0.853**	**0.823**	**0.903**	**0.715**	**0.834**	**0.778**	**0.875**	**0.796**	**0.886**	**0.821**	**0.901**

Click-based material selection.

```
Can you segment all pixels with the same
material where the <COLOR> star is located?
[MATERIAL_PROMPT]
```

Text-based material selection.

```
Segment every region that has the material
described below. [MATERIAL_PROMPT]
Description: <DESCRIPTION>
```

VQA.

```
Which of the following options best describes
the material where the <COLOR> star is located?
[MATERIAL_PROMPT]
```

Answer templates.

```
Sure, the segmentation result is [SEG].
```

Material prompt.

```
Regions with same base material but different
colors are considered as different materials.
However, regions with different lighting, shading
or shadows are considered as the same material.
```

<COLOR> is replaced with the star overlay color (e.g., red, cyan), and <DESCRIPTION> is replaced with the material description.

System prompt for object selection. For RefCOCO datasets, we follow the original implementation. The dataset contains short object descriptions, and the full question is formatted as:

```
What is the <DESCRIPTION> in this image?
Please output segmentation mask.
```

where <DESCRIPTION> is replaced with the referring expression (e.g., "left side monitor").

For EntitySeg, each datapoint contains a class name for the corresponding mask. To formulate instance segmentation, we provide the spatial location via a star overlay:

```
Can you segment the <CLASS_NAME> that contains
the <COLOR> star?
```

where <CLASS_NAME> is replaced with the object (e.g., chair, person).

In our experiments, we find that fine-tuning the entire VLM achieves significantly higher performance than using a low-rank adapter (LoRA). This is likely because our training data includes rich, fine-grained material descriptions, requiring the model to significantly adapt its visual-language representations.

3 Full Evaluation Results

In this section, we list all validation results that have been deferred due to limited space. Throughout, MAOAM refers to our model trained from Sa2VA model for 10 epochs on material and object data. We report results on material selection and object selection with two distinct prompting modalities: text- and click-based, and Visual Question Answering (VQA), when applicable.

Material-Centric Understanding. Tables 2 and 3 in the main text provide a comprehensive evaluation on material datasets: RealMat, SynMat, and SAMa. We evaluate material segmentation from click- and text-prompts, as well as reasoning via two VQA question types (Q1: sampling-based; Q2: hard-negative mining). MAOAM substantially outperforms existing models such as GLaMM and Sa2VA, which struggle with fine-grained material properties.

Object-Centric Grounding. To ensure that our material-specific tuning does not degrade general-purpose capabilities, we provide full results on standard benchmarks in Table 1. This includes the validation and test splits for RefCOCO, RefCOCO+, and RefCOCOg, alongside click-based selection on EntitySeg. The results indicate that MAOAM not only preserves but often improves upon the performance of the base Sa2VA model in traditional referring expression segmentation tasks.

4 Discussion and Ablation Studies

In this section, we perform further discussion and ablation studies that could not be included in the main draft due to limited space. All models used in ablation studies are initialized from GLaMM checkpoints and trained for 10 epochs, unless mentioned otherwise.

LoRA vs. standard fine-tuning. While the default configuration of GLaMM utilizes LoRA with rank 8 and alpha 16, our experiments indicate that LoRA yields significantly lower performance compared to standard fine-tuning on text-based selection tasks, as shown in Table 2. Interestingly, the LoRA-trained model demonstrates comparable or superior metrics in click-based selection. This suggests that low-rank adaptation is sufficient for processing local spatial information to produce accurate masks. However, standard fine-tuning of the LLM is clearly advantageous to interpret intricate and long material descriptions and align them with visual features. This performance gap is the most evident in VQA, where the model must reason within the text space to distinguish correct material attributes. For this reason, we follow standard fine-tuning as the default training strategy for GLaMM. For Sa2VA training, we follow the default setting (LoRA rank of 128) due to VRAM requirements.

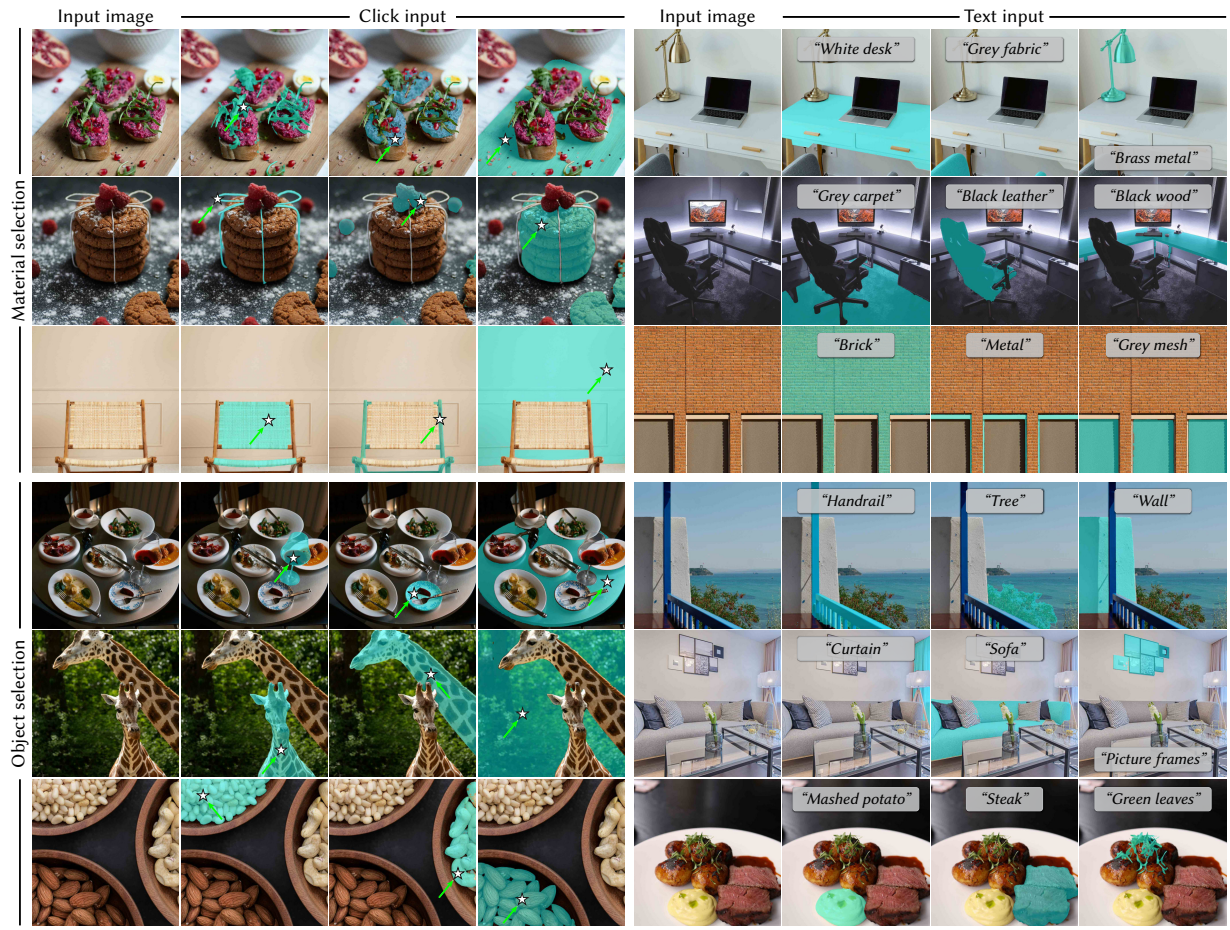


Fig. 2. Additional examples of our method on material selection (first three rows) and object selection (last three rows). We show both click-based queries (first four columns) and prompt-based queries (last four columns).

Table 2. Comparison of LoRA and full VLM fine-tuning for material understanding. We evaluate the performance across text-based selection, click-based selection, and VQA. Full VLM fine-tuning consistently outperforms the LoRA adaptation across all text-based tasks, including selection and VQA, while LoRA adaptation demonstrates comparable, and in certain cases better metrics on click-based selection.

	Material (Text-based Selection)						Material (Click-based Selection)						Material (VQA)					
	RealMat		SynMat		SAMa		RealMat		SynMat		SAMa		RealMat (Acc ↑)		SynMat (Acc ↑)		SAMa (Acc ↑)	
Method	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	Q1	Q2	Q1	Q2	Q1	Q2
LoRA	0.614	0.690	0.550	0.627	0.601	0.682	0.694	0.774	0.682	0.769	0.711	0.794	0.732	0.649	0.688	0.635	0.552	0.364
VLM	**0.670**	**0.739**	**0.582**	**0.658**	**0.661**	**0.739**	**0.760**	**0.830**	**0.726**	**0.806**	**0.756**	**0.832**	**0.818**	**0.976**	**0.775**	**0.983**	**0.693**	**0.847**

Effect of multi-task training. We evaluate the impact of our multi-task objective on material understanding by ablating three training configurations: (i) Click-only, which trains the model only on click-based selection; (ii) Click+Text, which combines click- and text-based selection; and (iii) All, our full multi-task framework with click-based selection, text-based selection, and VQA. All three models have been trained exclusively on material datasets.

As shown in Table 3, the results demonstrate a clear synergistic effect across tasks. The Click-only baseline performs well on spatial localization but cannot generalize to text-based queries. Adding text-based selection (Click+Text) restores referring performance, but the full multi-task configuration provides the largest gains. Specifically, including VQA not only complements text-based selection but also improves click-based selection, achieving the highest mIoU on RealMat over the three models. The varied results on click-based selection suggest that the three task formulations provide complementary supervision, resulting in a more robust model.

GLaMM vs Sa2VA. We compare two backbone configurations: GLaMM (LLaVA-v1.5 + SAM) and Sa2VA (Qwen2.5-VL-7B + SAM-2) after training on our material and object data. Specifically, we

Table 3. Ablation study on multi-task training. We compare three configurations: *Click*, *Click+Text*, and *All* (our multi-task training). We report grounding performance (mIoU and F1) and VQA Accuracy across all material datasets. We verify that introducing VQA questions helps improve the metrics on text-based selection task. The mixed results in click-based selection also signal that the three different task formulations complement each other. Note that VQA performance on *Click* and *Click+Text* models could not be measured because the models are not trained for VQA tasks. All models are trained on material data only, for 10 epochs.

	Material (Text-based Selection)						**Material (Click-based Selection)**						**Material (VQA)**					
	RealMat		SynMat		SAMa		RealMat		SynMat		SAMa		RealMat (Acc ↑)		SynMat (Acc ↑)		SAMa (Acc ↑)	
Strategy	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	Q1	Q2	Q1	Q2	Q1	Q2
Click	0.093	0.120	0.067	0.089	0.265	0.323	0.757	0.829	**0.735**	**0.814**	0.745	0.822	–	–	–	–	–	–
Click + Text	0.670	0.738	**0.589**	**0.666**	**0.655**	0.730	0.754	0.825	0.723	0.803	0.729	0.807	–	–	–	–	–	–
All	**0.675**	**0.746**	0.588	0.665	0.654	**0.730**	**0.756**	**0.828**	0.730	0.810	**0.730**	**0.809**	**0.833**	**0.970**	**0.771**	**0.979**	**0.705**	**0.835**

Table 4. Comparison between GLaMM and Sa2VA models' performance on material datasets, after being trained on our material and object dataset. Sa2VA (MAOAM) outperforms GLaMM by a large margin on both text- and click-based selection, despite being trained for fewer epochs. The two models show comparable performance on VQA.

	Material (Text-based Selection)						**Material (Click-based Selection)**						**Material (VQA)**					
	RealMat		SynMat		SAMa		RealMat		SynMat		SAMa		RealMat (Acc ↑)		SynMat (Acc ↑)		SAMa (Acc ↑)	
Method	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	Q1	Q2	Q1	Q2	Q1	Q2
GLaMM	0.670	0.739	0.582	0.658	0.661	0.739	0.760	0.830	0.726	0.806	**0.756**	**0.832**	0.818	**0.976**	0.775	**0.983**	0.693	0.847
Sa2VA (MAOAM)	**0.740**	**0.798**	**0.608**	**0.669**	**0.685**	**0.754**	**0.808**	**0.868**	**0.766**	**0.835**	0.747	0.823	**0.858**	0.974	**0.795**	0.979	**0.749**	**0.858**

Table 5. Comparison between GLaMM and Sa2VA models' performance on object datasets, after being trained on our material and object dataset. Sa2VA (MAOAM) outperforms GLaMM by a large margin across all text- and click-based object selection, despite being trained for less amount of epochs.

	RefCOCO (Text)						**RefCOCO+ (Text)**						**RefCOCOg (Text)**				**EntitySeg (Click)**	
	Val		TestA		TestB		Val		TestA		TestB		Val		Test		Val	
Method	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑	mIoU ↑	F1 ↑
GLaMM	0.772	0.834	0.798	0.857	0.754	0.819	0.719	0.778	0.761	0.819	0.677	0.740	0.727	0.792	0.734	0.799	0.777	0.846
Sa2VA (MAOAM)	**0.809**	**0.895**	**0.835**	**0.910**	**0.770**	**0.870**	**0.744**	**0.853**	**0.823**	**0.903**	**0.715**	**0.834**	**0.778**	**0.875**	**0.796**	**0.886**	**0.821**	**0.901**

train GLaMM for 15 epochs and Sa2VA for 10 epochs, resulting in comparable wall-clock time.

Table 4 and Table 5 report performance on material and object datasets, respectively. Sa2VA (MAOAM) substantially outperforms GLaMM across text- and click-based interactions, on both material and object selection, despite being trained for fewer epochs. VQA performance is comparable between the two models, with GLaMM slightly outperforming on some splits and MAOAM on others.

These results suggest that the more recent VLM backbone (Qwen2.5-VL) can better align complex text queries with visual-semantic representations that benefit both material and object selection. We therefore use Sa2VA as our primary model (MAOAM) but mainly use GLaMM for ablation studies due to lower computational cost.

We note that the Sa2VA-based variant yields higher quantitative metrics, while the GLaMM-based variant generalizes better and is more robust during inference. Hence, quantitative results are from the former and qualitative examples from the latter.

Data scaling. We further evaluate the scalability of our data generation framework by varying both the amount of training data and the number of training epochs. As shown in Table 6, training with only 50% of randomly sampled material data remains competitive with full-scale training across all three material benchmarks. In Table 7, we also report performance after 5, 10, and 15 epochs of material training. While performance improves with longer training, 5–10 epochs of training already achieve competitive results. Together Table 4 in the main paper, these results suggest practical flexibility in the data generation and training pipeline, depending on the available compute budget.

Table 6. Data scale analysis. We report mIoU for text- and click-based material selection when training with the full material dataset and a 50% randomly subsampled version. Half-scale training remains competitive with full-scale training across all three material benchmarks.

	RealMat		SynMat		SAMa	
Training data	Text	Click	Text	Click	Text	Click
Half	0.641	0.732	0.570	0.726	0.639	0.727
Full	**0.675**	**0.756**	**0.588**	**0.730**	**0.654**	**0.730**

Table 7. Training epoch analysis. We report mIoU for text- and click-based material selection after 5, 10, and 15 epochs of training. Performance improves with longer training, while 5–10 epochs provide competitive results.

	RealMat		SynMat		SAMa	
Training	Text	Click	Text	Click	Text	Click
5 epochs	0.640	0.717	0.565	0.702	0.625	0.675
10 epochs	0.656	0.741	0.586	0.725	**0.662**	0.713
15 epochs	**0.675**	**0.756**	**0.588**	**0.730**	0.654	**0.730**

5 Further Applications in Medical Imaging Data

The SurgVu24 [Organizers 2024] challenge released a medical image dataset for classifying and localizing different surgical tools. To demonstrate the practical usefulness of our model beyond image editing tasks, we evaluate whether MAOAM generalizes to out-of-domain images. Fig. 3 shows click-based object selection on surgical imagery. Despite never being trained on medical data, our model produces pixel-accurate masks for surgical tools with simple click interactions, suggesting that the visual grounding learned from our material and object training transfers to novel domains.

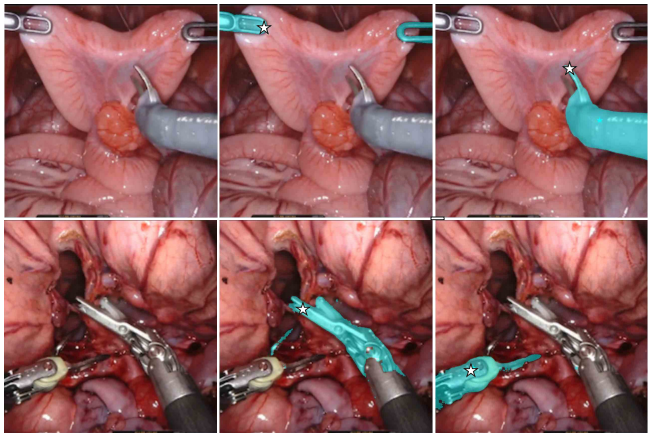


Fig. 3. Material selection on medical images. We show that our model generalizes well to extremely out-of-domain examples, such as medical imagery, and that our model is able to output pixel-level accurate masks with simple click operations.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. 2025. Qwen2.5-VL Technical Report. *arXiv preprint arXiv:2502.13923* (2025).

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. https://lmsys.org/blog/2023-03-30-vicuna/

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models.. In *ICLR*. OpenReview.net. http://dblp.uni-trier.de/db/conf/iclr/iclr2022.html#HuSWALWWC22

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*. 4015–4026.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Computer Vision – ECCV 2014*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer International Publishing, Cham, 740–755.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 26296–26306.

SurgVu24 Challenge Organizers. 2024. SurgVu24: Surgical Visual Understanding Challenge. https://surgvu24.grand-challenge.org/. Part of the EndoVis Challenge at MICCAI 2024.

Lu Qi, Jason Kuen, Tiancheng Shen, Jiuxiang Gu, Weidong Guo, Jiaya Jia, Zhe Lin, and Ming-Hsuan Yang. 2023. High-Quality Entity Segmentation. In *International Conference on Computer Vision (ICCV)*.

Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S Khan. 2024. Glamm: Pixel grounding large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13009–13018.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. 2024. SAM 2: Segment Anything in Images and Videos. *arXiv preprint arXiv:2408.00714* (2024). https://arxiv.org/abs/2408.00714

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. 2021. LAION-400M: Open Dataset of CLIP-Filtered 400 Million Image-Text Pairs. arXiv:2111.02114 [cs.CV] https://arxiv.org/abs/2111.02114

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2017. Scene Parsing Through ADE20K Dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.