# E-SAM: Training-Free Segment Every Entity Model

Weiming Zhang[1] Dingwen Xiao[1] Lei Chen[1,2] Lin Wang[3†]
[1] HKUST (GZ) [2] HKUST [3] Nanyang Technological University
zweiming996@gmail.com, dxiaoaf@connect.hkust-gz.edu.cn, leichen@cse.ust.hk, linwang@ntu.edu.sg
Project Page: https://weimingz996.github.io/E-SAM/

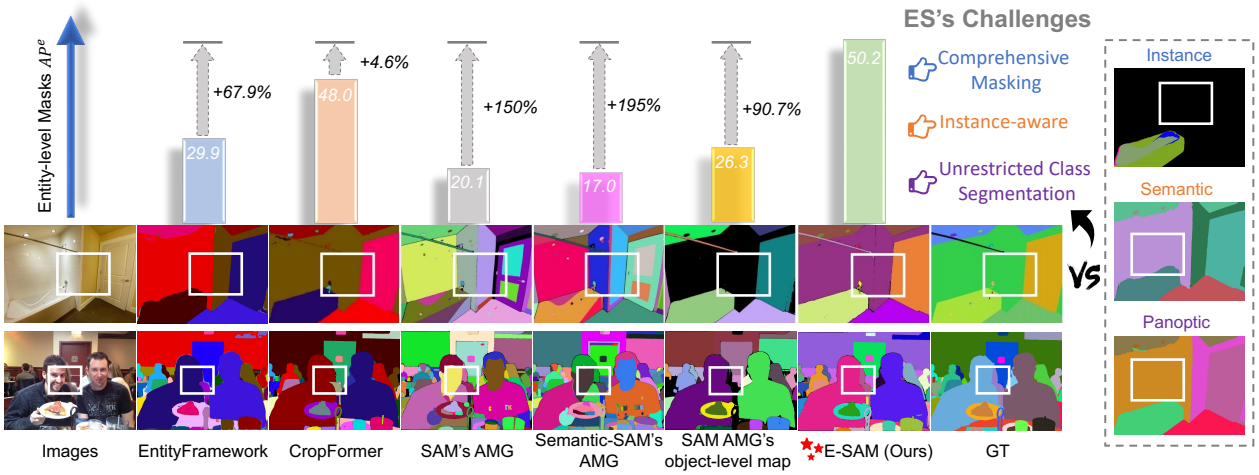


Figure 1. Compared with prior ES methods [29, 30], SAM and Semantic-SAM [21], our E-SAM demonstrates superior performance. Notably, our E-SAM, without any additional training, significantly outperforms both SAM's AMG and its object-level map.

## Abstract

*Entity Segmentation (ES) aims at identifying and segmenting distinct entities within an image without the need for predefined class labels. This characteristic makes ES well-suited to open-world applications with adaptation to diverse and dynamically changing environments, where new and previously unseen entities may appear frequently. Existing ES methods either require large annotated datasets or high training costs, limiting their scalability and adaptability. Recently, the Segment Anything Model (SAM), especially in its Automatic Mask Generation (AMG) mode, has shown potential for holistic image segmentation. However, it struggles with over-segmentation and under-segmentation, making it less effective for ES. In this paper, we introduce* ***E-SAM,*** *a novel training-free framework that exhibits exceptional ES capability. Specifically, we first propose* ***Multi-level Mask Generation (MMG)*** *that hierarchically processes SAM's AMG outputs to generate reliable object-level masks while preserving fine details at other levels.* ***Entity-level Mask Refinement (EMR)*** *then refines these object-level masks into accurate entity-level masks. That is, it separates overlapping masks to address the redundancy issues inherent in SAM's outputs and merges similar masks by evaluating entity-level consistency. Lastly,* ***Under-Segmentation Refinement (USR)*** *addresses under-segmentation by generating additional high-confidence masks fused with EMR outputs to produce the final ES map. These three modules are seamlessly optimized to achieve the best ES without additional training overhead. Extensive experiments demonstrate that E-SAM achieves state-of-the-art performance compared to prior ES methods, demonstrating a significant improvement by* ***+30.1*** *on benchmark metrics.*

## 1. Introduction

Entity Segmentation (ES) [30] is an emerging task in computer vision that focuses on segmenting visual entities in an image without relying on predefined class labels. Unlike traditional segmentation tasks, which are limited by fixed categories, ES aligns more closely with human perception [26], where entities are identified based on visual coherence. As shown in Figs. 1 & 2, ES offers *comprehensive masking* in instance segmentation, enhances the *instance awareness* of semantic segmentation, and overcomes the *label class limitations* in panoptic segmentation, where certain undefined entities (e.g., the metal rod in Fig. 1 and exhaust fan in Fig. 2) are failed to be segmented. This

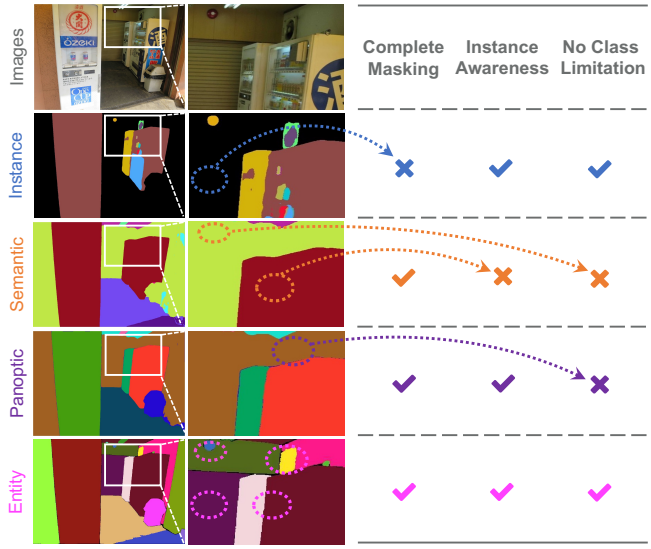


Figure 2. Comparison of ES and with three segmentation tasks.

class-agnostic nature makes ES well-suited to various open-world applications, from image editing [22] and manipulation [35, 37] to real-time environments such as autonomous driving [8, 44], robotics [1, 6, 39], and surveillance [9, 11]. However, existing ES methods [2, 29–31, 36] often face high training costs and require extensive annotated datasets, limiting their scalability and practicality. In addition, these models struggle with generalization in diverse scenarios.

The Segment Anything Model (SAM) [18], as a foundational model for segmentation, demonstrates robust zero-shot performance due to its training on a vast dataset containing over one billion masks. SAM can accept various prompts, such as points, boxes, and masks, to segment target entities and allows for the continuous refinement of segmentation results through additional prompts. Furthermore, SAM's Automatic Mask Generation (AMG) mode was specifically designed to perform full-image instance segmentation without the need for explicit prompts [21, 34, 40]. However, due to AMG's inherent limitations, its performance for ES is suboptimal. Specifically, the uniform point prompt sampling strategy in AMG enables SAM to generate three mask levels (object, part, sub-part) for each point. However, applying a naive Non-Maximum Suppression (NMS) [28] to remove redundant masks often results in over-segmentation and under-segmentation, either discarding fine details or failing to remove unnecessary overlaps, see Fig. 1. To overcome these limitations, we explore a novel question: *How can we efficiently and effectively achieve ES of all entities in an image?*

To this end, we propose **E-SAM**, a novel *training-free* framework specifically designed the achieve state-of-the-art ES performance without incurring additional training costs. Our E-SAM effectively mitigates the over-segmentation and under-segmentation challenges inherent in AMG by integrating three key modules: Multi-level Mask Generation (MMG) (Sec.3.2), Entity-level Mask Refinement (EMR) (Sec.3.3), and Under-Segmentation Refinement (USR) (Sec.3.4). Specifically, the MMG module first categorizes the AMG's outputs according to their area levels and confidence scores, subsequently applying diverse NMS strategies with varying thresholds to retain high-confidence object-level masks while preserving additional masks at the part and subpart levels in densely populated regions. Then, the EMR module begins by increasing the number of uniformly sampled points to construct a mask gallery, which includes a diverse set of high-confidence masks. Next, this gallery is utilized to identify and separate overlapping object-level masks, refining them into distinct adjacent masks to eliminate redundancy. Subsequently, EMR constructs a similarity matrix among these refined masks and leverages the mask gallery to merge highly similar masks, ultimately producing an accurate entity-level segmentation map. Lastly, the USR module refines under-segmented regions in EMR's outputs by incorporating superpixel centroids, as well as part and subpart mask centroids, as prompts to guide additional segmentation. By seamlessly integrating the three modules, our E-SAM efficiently generates high-quality ES masks for the entire image without incurring additional training overhead.

We conducted extensive experiments that demonstrated the effectiveness of our framework across multiple datasets. As shown in Fig. 1, our E-SAM consistently outperforms the previous state-of-the-art ES methods [29, 30] and SAM. In particular, under the same backbone size, our E-SAM consistently outperformed SAM's AMG by more than double in performance according to benchmark metrics. As illustrated in Fig. 7, our E-SAM exhibits strong generalization capability even in unseen datasets (or open-world scenarios), underscoring the novelty and effectiveness of our E-SAM design, particularly in its training-free approach.

In summary, our contributions are as follows: **(I)** We introduce E-SAM, a novel **training-free** framework aimed at enhancing SAM's performance in entity segmentation. **(II)** We design three modules, MMG, EMR, and USR, that work in sequence to generate reliable masks at multiple levels, eliminate overlapping regions, refine masks at the entity level, and optimize undersegmented areas, ultimately producing a high-quality entity-level segmentation map. **(III)** Extensive experiments demonstrate that E-SAM significantly outperforms both SAM and existing ES methods, achieving superior performance across diverse datasets with strong quantitative and qualitative results.

## 2. Related Works

**Segment Anything Model (SAM) and AMG.** SAM [18] is a foundation model trained on a large-scale dataset containing approximately 1 billion masks, providing robust zero-

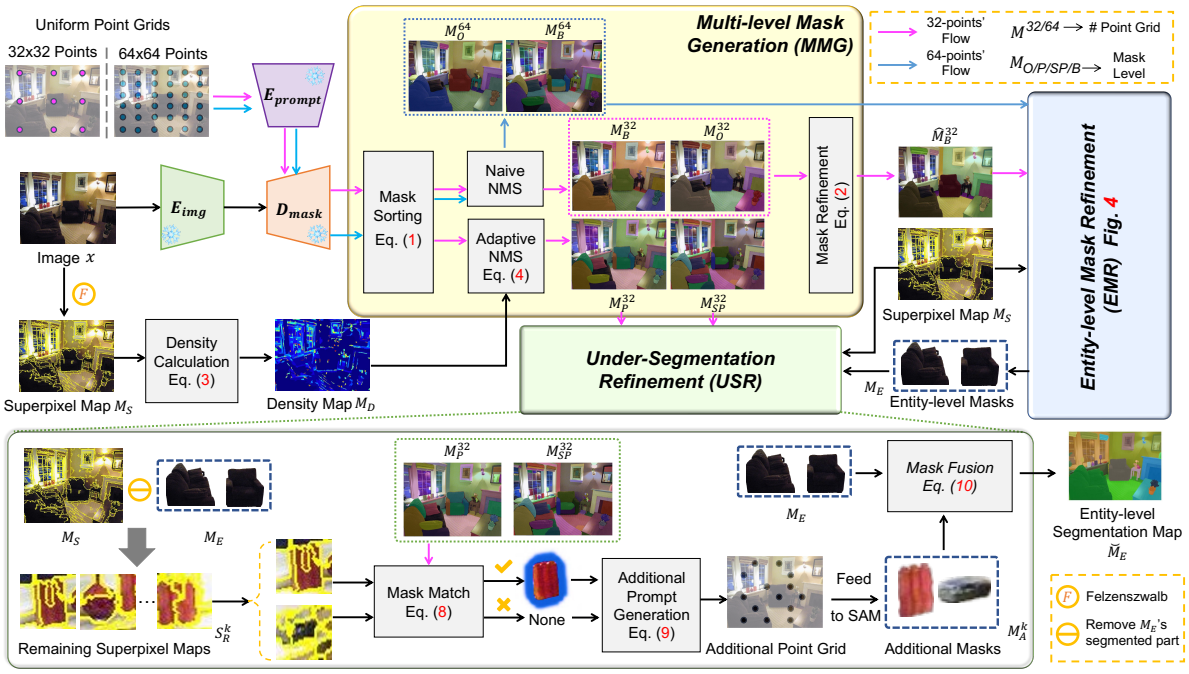


Figure 3. Overview of our E-SAM framework. Our method contains three main technical components: Multi-level Mask Generation (MMG), Entity-level Mask Refinement (EMR), and Under-Segmentation Refinement (USR).

shot capabilities in segmentation tasks. SAM accepts visual prompts such as points, bounding boxes, or masks, enabling interactive segmentation. This versatility has allowed SAM to find applications in a wide range of domains, including object tracking [5, 32, 41, 46], 3D instance segmentation [3, 42, 43], and medical imaging [15, 25, 27, 45]. In addition to its interactive capabilities, SAM includes an Automatic Mask Generation (AMG) mode, which is designed to segment everything within a given image without requiring explicit prompts. SAM's AMG has been explored extensively in recent studies [10, 17, 20, 34, 40] to further extend the capabilities of SAM in various domains. Despite the successes of SAM's AMG, it is not without challenges. The uniform point sampling strategy in AMG returns three masks per point, often leading to redundancy and overlapping regions. Using a basic Non-Maximum Suppression (NMS) [28] to filter these masks may inadvertently remove crucial object parts(leading to under-segmentation) or retain too many similar masks (causing over-segmentation). *To address these shortcomings, we propose three modules—MMG, EMR, and USR—that mitigate the issues of over-segmentation and under-segmentation, enabling efficient and accurate segmentation of each entity.*

**Entity Segmentation.** Entity Segmentation (ES) is a novel task that was first introduced in [30]. The core objective of ES is to segment all perceptually distinct entities within an image, irrespective of any predefined class labels. Several approaches [2, 29–31, 36] have been proposed to solve ES, including methods like CropFormer [29], which introduced the EntitySeg dataset with high-quality, densely annotated masks to enhance generalization for segmentation models. However, these specialized models have certain weaknesses. These methods rely heavily on costly, labor-intensive annotated datasets, limiting their scalability and applicability, and hindering real-world generalization. Additionally, training these models requires significant computational resources, adding complexity and limiting broader adoption. *To address these challenges, our E-SAM optimizes SAM's AMG through a training-free approach to efficiently enhance its performance and achieve high-quality entity-level segmentation maps.*

## 3. Methodology

### 3.1. Overview

In this section, we first provide an overview of our framework – **E-SAM**, shown in Fig. 3. E-SAM operates training-free, eliminating the need for training data or training costs. Therefore, we freeze SAM's image encoder $E_{\text{img}}$, prompt encoder $E_{\text{prompt}}$, and mask decoder $D_{\text{mask}}$. Given a test image $x \in \mathbb{R}^{H \times W \times 3}$, E-SAM follows the SAM's AMG processes [18] by inputting the image $x$ into the image encoder $E_{\text{img}}$. E-SAM then uniformly generates point prompts along each side, which are fed into the prompt encoder $E_{\text{prompt}}$. In the subsequent mask decoding stage $D_{\text{mask}}$, E-SAM generates and selects high-confidence segmentation masks across the levels of the object, part, and subpart for each point prompt. Since E-SAM generates masks of multiple granularities, the main challenges of E-SAM lie in: (1)

effectively removing overlapping masks, (2) fusing SAM-generated masks to obtain precise entity-level masks, and (3) refining under-segmented regions to ensure complete segmentation of all entities. To address these challenges, our E-SAM consists of three key modules: Multi-level Mask Generation (MMG) Module (Sec. 3.2), Entity-level Mask Refinement (EMR) Module (Sec. 3.3), and Under-Segmentation Refinement (USR) Module (Sec. 3.4). We now describe these modules in detail.

### 3.2. Multi-level Mask Generation (MMG)

The MMG module aims to stratify the outputs of SAM into different levels by employing a combination of NMS methods and thresholds, ultimately generating distinct mask maps to mitigate the over-segmentation issues caused by multi-granularity masks. First, for a given image $x$, MMG uniformly generates 32-point prompts per side. Then, MMG categorizes the masks returned by SAM for each point prompt based on their area into three levels: object-level mask map $M_O^{32}$, part-level mask map $M_P^{32}$, and subpart-level mask map $M_{SP}^{32}$. Additionally, MMG selects the mask with the highest confidence score for each point to form the best-level mask map $M_B^{32}$, as shown in Eq. (1):

$$\begin{aligned} M_i^{32} &= \{M_{i,O}^{32}, M_{i,P}^{32}, M_{i,SP}^{32}\}, \quad A_{i,O} \geq A_{i,P} \geq A_{i,SP}, \\ \varepsilon_{i,B} &= \underset{\varepsilon \in \{O,P,SP\}}{\arg\max}\ s_{i,\varepsilon}, \quad M_{i,B}^{32} = M_{i,\varepsilon_{i,B}}^{32}, \end{aligned} \tag{1}$$

where $i$ denotes the a point prompt, $A_{i,\varepsilon}$ represents the area and $s_{i,\varepsilon}$ indicates the confidence score of the $\varepsilon$-th level mask returned for point $i$. Furthermore, for each mask level, MMG applies specific strategies: **Object-Level Mask Map:** Given that $M_{i,O}^{32}$ best approximates entity-level masks, MMG retains only high-confidence object-level masks, thereby reducing fragile noise in SAM's object-level predictions. To achieve this, MMG first applies naive NMS with a high threshold, denoted as $\theta_O$. The remaining object-level masks $M_{i,O}^{32}$ are then checked for presence within the best-level mask map $M_B^{32}$ based on Intersection over Union (IoU) criteria. Masks that do not sufficiently match are discarded, resulting in the refined object-level mask map $\hat{M}_O^{32}$.

$$\max\ \text{IoU}(M_{i,O}^{32}, M_{*,B}^{32}) \geq \gamma_O \implies M_{i,O}^{32} \in \hat{M}_O^{32}, \tag{2}$$

where $\gamma_O$ denotes the IoU threshold. **Part-Level and Subpart-Level Mask Maps:** MMG aims to generate additional masks in complex regions, capturing finer details of densely populated entities. To facilitate this, MMG first employs Felzenszwalb's method [7] for superpixel clustering on $x$, which generates the superpixel map $M_S$, with superpixels denoted as $S_k$. Then, MMG creates a density map $M_D$ for $x$, indicating entity density across regions.

$$M_D(x) = \sum_{k=1}^{K} w_k \cdot \mathbf{1}_{S_k}(x), \tag{3}$$

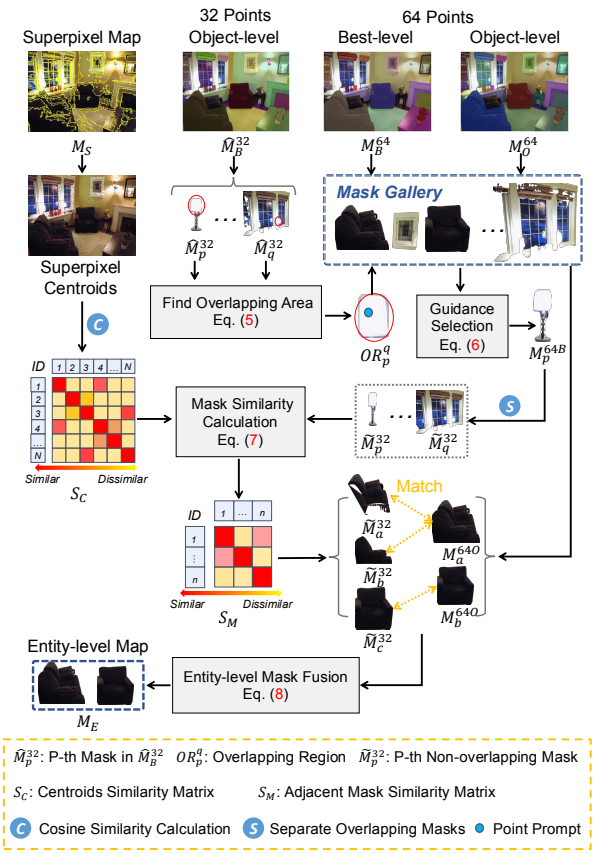


Figure 4. Overview of our EMR framework.

where $K$ denotes the number of superpixels and $w_k$ denotes the weight assigned to the superpixel. $\mathbf{1}_{S_k}(x)$ is an indicator function whether pixel belongs to $S_k$. MMG then uses $M_D$ in conjunction with Adaptive NMS [24] to dynamically adjust the threshold $N_{\mathcal{M}}$ in densely populated regions.

$$N_{\mathcal{M}} := \max(N_t, M_D(\mathcal{M})), \tag{4}$$

where $N_t$ is a base threshold and $\mathcal{M}$ denotes the candidate mask. Given SAM's robust zero-shot capabilities, precise point prompts allow SAM to provide satisfactory masks for target entities. Thus, MMG also increases the number of point prompts per side to 64 and re-feeds these to $E_{\text{prompt}}$ to obtain a richer object-level mask map $M_O^{64}$ and a best-level mask map $M_B^{64}$ for subsequent refinement.

### 3.3. Entity-level Mask Refinement (EMR)

The EMR module, illustrated in Fig. 4, aims to refine $\hat{M}_O^{32}$ into the entity-level mask map $M_E$ using the outputs of MMG: $M_O^{64}$ and $M_B^{64}$. Specifically, EMR aggregates $M_O^{64}$ and $M_B^{64}$ into a mask gallery $G$, which acts as a reference for subsequent refinement. For the masks in $\hat{M}_O^{32}$, we employ a "split-then-merge" strategy to achieve refinement. First, all masks are sorted in descending order based on their predicted scores. For each mask in this sorted list, denoted as $\hat{M}_p^{32}$, EMR identifies overlapping masks $\hat{M}_q^{32}$

(where $p \neq q$) and calculates the overlap region $OR_p^q$.

$$OR_p^q = M_p^{32} \cap M_q^{32}, \quad \forall p \neq q. \tag{5}$$

If the area of $OR_p^q$ relative to the largest mask between $\hat{M}_p^{32}$ and $\hat{M}_q^{32}$ is below a threshold $\delta$, the overlapping region in the larger mask is removed. Otherwise, EMR finds the existing point prompts from the 64-points-per-side sampling strategy within $OR_p^q$, denoted as $P^{64}$. Next, the EMR module considers the masks in the mask gallery corresponding to $P^{64}$: the object-level mask $M_p^{64O}$ and the best-level mask $M_p^{64B}$. We set a tolerance level $\tau$: if the score difference between $M_p^{64O}$ and $M_p^{64B}$ is within $\tau$, $M_p^{64O}$ is selected as the guidance for refining the overlap region. Otherwise, $M_p^{64B}$ is used. When multiple prompts are present, the mask that appears most frequently is chosen for guidance.

$$G_p = \begin{cases} M_p^{64O}, & \text{if } S_p^{64B} - S_p^{64O} < \tau, \\ M_p^{64B}, & \text{if } S_p^{64B} - S_p^{64O} \geq \tau. \end{cases} \tag{6}$$

Subsequently, using the selected guidance mask $G_p$, EMR updates the $\hat{M}_p^{32}$ and $\hat{M}_q^{32}$, resulting in non-overlapping masks $\tilde{M}_p^{32}$ and $\tilde{M}_q^{32}$. Next, based on the image features extracted by $E_{\text{img}}$, we construct a centroids cosine similarity matrix $S_C$ for the centroids in the $M_S$. Then, the EMR module identifies the superpixel centroids present within each mask and determines which masks their most similar corresponding centroids belong to. By analyzing the frequency with which each mask appears, we construct the adjacent mask similarity matrix $S_M$.

$$S_M(i,j) = \frac{1}{|C_i|} \sum_{c_i \in C_i} \left| \{ c_j \mid c_j \in C_j \wedge c_j \in \text{Top}_k(S_C(c_i, \cdot)) \} \right|, \tag{7}$$

where $C_i$ and $C_j$ denotes the set of superpixel centroids in mask $i$ and $j$, and $|\cdot|$ marks the cardinality. $\text{Top}_k(S_C(c_i, \cdot))$ represents the top $k$ most similar centroids to centroid $c_i$, based on similarity scores from $S_C$. Subsequently, the EMR module evaluates the masks with high similarity scores. It checks the Mask Gallery $G$ to determine if there exists a mask that encompasses both of the highly similar masks. When masks match, they are merged into a unified entity-level mask; otherwise, they remain separate.

$$M_E = \begin{cases} M_a^{64} \cup M_b^{64}, & \text{if } M_a^{64}, M_b^{64} \in G \text{ and match exists} \\ \{M_a^{32}, M_b^{32}\}, & \text{otherwise} \end{cases} \tag{8}$$

After matching and fusing adjacent masks, the EMR produces the entity-level map $M_E$.

### 3.4. Under-Segmentation Refinement (USR)

The USR module, as illustrated in Fig. 3, aims to address the under-segmentation issues that may arise from the initial outputs of MMG. The module further refines $M_E$ to ensure comprehensive coverage of all perceptually distinct entities. The USR first considers the regions in the superpixel map $M_S$ that are not covered by the entity-level map $M_E$ and divides them into the remaining superpixel maps $S_R^k$, where $k$ represents the number of maps. Then, for each $S_R^i$, different cases are considered. When $S_R^i$ is contained within a mask from $M_P^{32}$ or $M_{SP}^{32}$, its centroid is used as an additional point prompt $P_A^i$ for SAM; otherwise, the superpixel's centroid is used. The resulting additional masks generated by USR are denoted as $M_A^k$.

$$P_A^i = \begin{cases} \text{centroid}(M_P^{32} \cup M_{SP}^{32}), & \text{if } S_R^i \subseteq (M_P^{32} \cup M_{SP}^{32},) \\ \text{centroid}(S_R^i), & \text{otherwise.} \end{cases} \tag{9}$$

Next, the USR module evaluates each mask in $M_A^k$ against the current entity-level map $M_E$. If the IoU between a mask in $M_A^k$ and an entity-level mask in $M_E$ exceeds a specified threshold $\rho$, the mask is retained as part of the entity. Otherwise, it is preserved as an independent entity-level mask. Consequently, $M_E$ is refined to $\breve{M}_E$.

$$\breve{M}_E^b = \begin{cases} M_E^a \cup M_A^a, & \text{if } iou(M_E^a, M_A^a) > \rho, \\ M_A^b, & \text{otherwise.} \end{cases} \tag{10}$$

To prevent the generation of an excessive number of part-level masks, a naive greedy algorithm is designed within the USR module. This means that USR attempts to use fewer masks from $M_A^k$ to fill the unsegmented regions in $M_E$, optimizing the overall segmentation quality. Through these refinements, USR significantly enhances the robustness and completeness of the segmentation, particularly in regions with complex boundaries or dense entity arrangements.

## 4. Experiments

### 4.1. Datasets and Implementation Details

**Datasets** We leverage the EntitySeg benchmark dataset (EntitySeg) [29] along with a panoptic segmentation dataset, COCO2017val [23], and a large-scale SA1B [18] dataset, to evaluate the performance of our E-SAM. *More dataset details can be found in the Supplement.*

**Implementation details** For the entire evaluation process, we utilized two NVIDIA A40 GPUs along with eight NVIDIA 3090 GPUs. We adopt the pre-trained SAM backbones, specifically ViT-B, ViT-L, and ViT-H. Moreover, we adopt the $AP^e$ metric for open-world entity segmentation.

### 4.2. Comparisons with Existing Works

In this work, we conducted experiments on the EntitySeg dataset using both high-resolution (HR) and low-resolution (LR) testing subsets. Tabs. 1 and 2 present the comparative results for various existing methods and different backbones. Fig. 5 demonstrates the visual comparisons. *More visualizations are available in Supplement.*

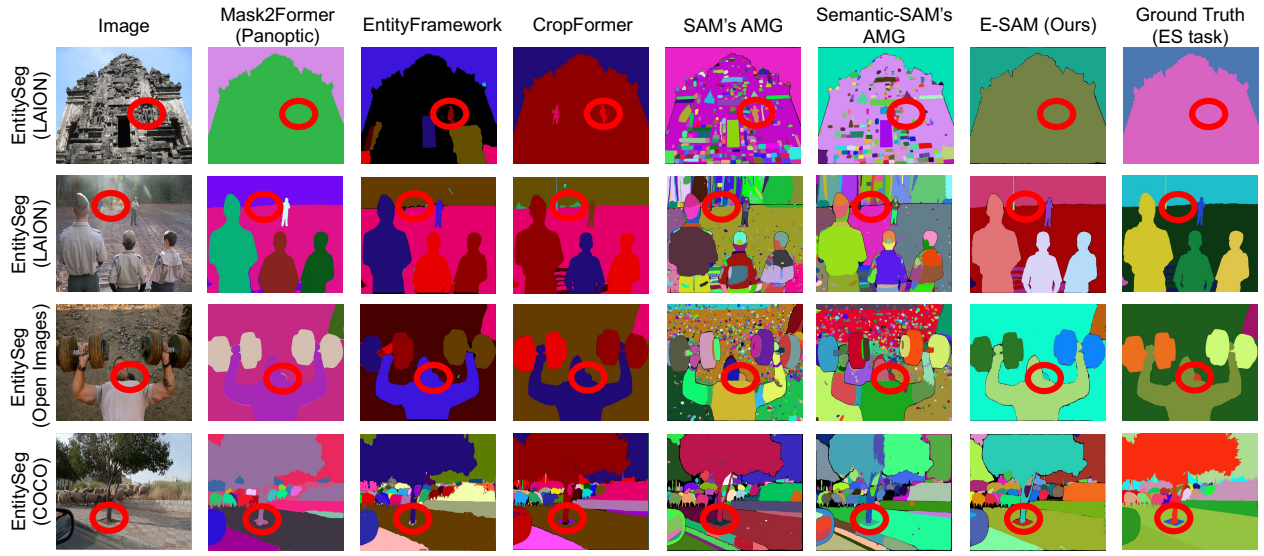


Figure 5. Example visualizations comparing various methods on different data sources in EntitySeg [30], including COCO [23], LAION [33], and Open Images [19]. The corresponding ground truth of the entity segmentation task is provided in the last column.

| | Methods | Backbone | EntitySeg | | |
|---|---|---|---|---|---|
| | | | $AP^e$ | $AP^e_{50}$ | $AP^e_{75}$ |
| Panoptic Methods | ○ Mask-RCNN [12] | Swin-T | 28.4 | 49.2 | 28.1 |
| | ○ Mask Transfiner [16] | Swin-T | 33.7 | - | - |
| | ○ PatchDCT [38] | Swin-T | 35.4 | - | - |
| | ○ Mask2Former [4] | Swin-T | 40.9 | 58.1 | 41.6 |
| | | Swin-L | 46.2 | 63.7 | 47.5 |
| ES Methods | ○ EntityFramework [30] | FPN | 29.9 | 47.6 | 30.1 |
| | ○ CropFormer [29] | Swin-T | 42.7 | 59.7 | 43.8 |
| | | Swin-L | 48.0 | 65.3 | 49.3 |
| SAM-based Methods | ○ SAM [18] | ViT-B | 13.7 | 19.1 | 13.4 |
| | | ViT-L | 19.7 | 28.8 | 18.9 |
| | | ViT-H | 20.1 | 32.9 | 19.4 |
| | ○ Semantic-SAM [21] | Swin-T | 16.1 | 22.7 | 16.4 |
| | | Swin-L | 17.0 | 24.6 | 17.2 |
| | ☆ E-SAM (Ours) | ViT-B | 40.1 | 57.8 | 38.9 |
| | | ViT-L | 45.9 | 62.7 | 45.3 |
| | | ViT-H | **50.2** | **66.8** | **49.9** |

Table 1. Comparison with prior methods ( panoptic-based, ES-based, and SAM-based ) on the EntitySeg benchmark.

| Method | Backbone | EntitySeg-LR ($AP^e_L$) |
|---|---|---|
| ○ Mask2Former [4] | Swin-T | 38.8 |
| | Swin-L | 44.4 |
| ○ CropFormer [29] | Swin-T | 40.6 |
| | Swin-L | 45.8 |
| ○ SAM [18] | ViT-B | 10.5 |
| | ViT-L | 17.3 |
| | ViT-H | 17.0 |
| ○ SAM$^O$ [18] | ViT-B | 12.2 |
| | ViT-L | 21.6 |
| | ViT-H | 23.7 |
| ☆ E-SAM (Ours) | ViT-B | 35.8 |
| | ViT-L | 43.6 |
| | ViT-H | **48.9** |

Table 2. Comparison of various entity segmentation approaches with different backbones on the EntitySeg-LR benchmark, evaluated in terms of $AP^e_L$. SAM$^O$ denotes object-level map.

**Comparison with Panoptic Methods.** Tab. 1 compares our E-SAM with several state-of-the-art (SOTA) panoptic segmentation methods, including Mask-RCNN [12], Mask Transfiner [16], PatchDCT [38] and Mask2Former [4]. Our E-SAM with ViT-B and ViT-L achieves comparable performance to Mask2Former. Furthermore, with the ViT-H backbone, our method outperforms all baselines, achieving $AP^e$ scores of **50.2**, surpassing Mask2Former's best performance. The results of the LR subset show that our E-SAM excels with a score of **48.9** when using the ViT-H backbone.

**Comparison with ES Methods** In the second group, we compare our E-SAM with the methods specifically designed for ES task, such as EntityFramework [30] and CropFormer [29]. These methods are tailored to entity segmentation, giving them an advantage with similarly sized backbones. Although our E-SAM shows slightly lower performance compared to CropFormer when using the ViT-B and ViT-L backbones. However, with larger backbones, such as ViT-H, our E-SAM achieves a notable improvement, reaching **50.2** $AP^e$ and **48.9** $AP^e_L$ without incurring additional training costs associated with larger model training.

**Comparison with SAM-based Methods** In the third comparison, we evaluate our E-SAM alongside other SAM-based methods, specifically SAM [18] and Semantic-SAM [21]. In the high-resolution setting, SAM (ViT-H)

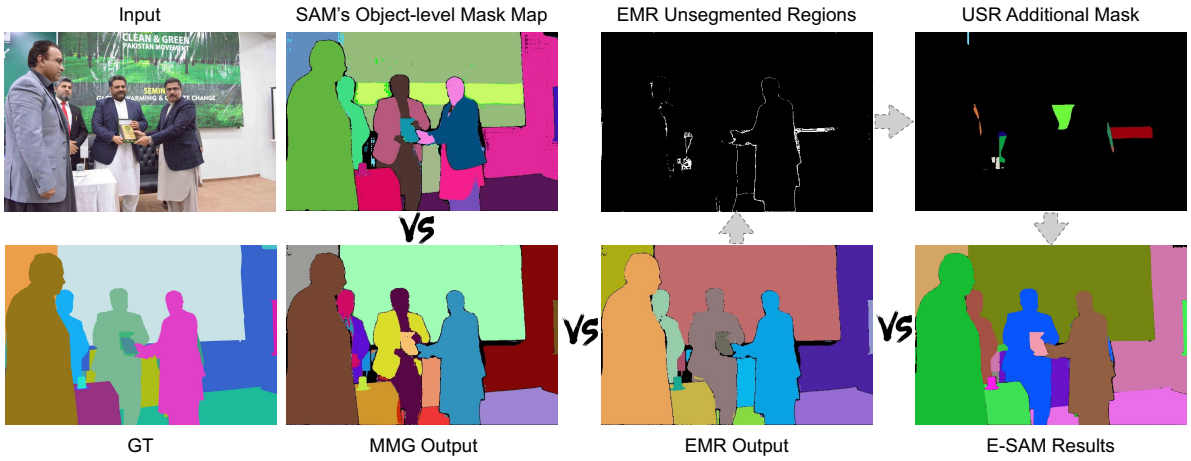


Figure 6. Example visualizations of the sequential application of the three key modules.

| MMG | EMR | USR | EntitySeg | | |
|---|---|---|---|---|---|
| | | | $AP_L^e$ | $AP_L^{e50}$ | $AP_L^{e75}$ |
| - | - | - | 17.3 | 23.1 | 18.0 |
| ✓ | - | - | 20.3 | 40.4 | 19.7 |
| - | ✓ | - | 29.5 | 50.6 | 29.1 |
| - | - | ✓ | 22.7 | 31.4 | 23.0 |
| - | ✓ | ✓ | 40.8 | 57.4 | 40.1 |
| ✓ | - | ✓ | 38.1 | 54.9 | 37.2 |
| ✓ | ✓ | - | 35.0 | 54.1 | 35.4 |
| ✓ | ✓ | ✓ | **43.6** | **60.8** | **43.1** |

Table 3. Ablation study on three key modules of E-SAM.

| $\theta_O$ | | $\gamma_O$ | | $\delta$ | |
|---|---|---|---|---|---|
| Value | $AP^e$ | Value | $AP^e$ | Value | $AP^e$ |
| 0.7 | 19.5 | 0.3 | 17.6 | 0.01 | 34.1 |
| 0.75 | 19.8 | 0.9 | 18.2 | 0.05 | **35.0** |
| 0.8 | **20.3** | 0.6 | **20.3** | 0.1 | 34.4 |
| 0.9 | 19.6 | 1.0 | 19.6 | 0.2 | 33.8 |
| $\tau$ | | $\rho$ | | Point Prompts | |
| Value | $AP^e$ | Value | $AP^e$ | Value | $AP^e$ |
| 0 | 34.1 | 0 | 43.4 | 16/16 | 39.8 |
| 0.05 | 34.5 | 0.1 | **43.6** | 16/32 | 41.9 |
| 0.1 | **35.0** | 0.3 | 42.9 | 16/64 | 42.5 |
| 0.2 | 34.2 | 0.5 | 42.6 | 32/64 | **43.6** |

Table 4. Ablation study on hyperparameters and the number of point prompts. All performance is evaluated in each module.

achieves only an $AP^e$ of **20.1**, whereas our E-SAM, utilizing the ViT-H backbone, significantly improves this score to **50.2** $AP^e$. Similarly, compared to Semantic-SAM (Swin-L), which scores **17.0** $AP^e$, our E-SAM with the ViT-L backbone surpasses it by a margin of **+28.9** $AP^e$. For low-resolution images, SAM (ViT-H) achieves an $AP_L^e$ of **23.7**, while our E-SAM with the ViT-H backbone attains **48.9** $AP_L^e$, improving the performance by **+25.2** $AP_L^e$.

## 4.3. Ablation Studies and Analysis

### 4.3.1. Ablation of Three Key Module

**Effectiveness of MMG Module.** Tab. 3 highlights the effectiveness of the MMG module based on ViT-L backbone and EntitySeg-LR test set. Adding MMG to the baseline (which consists solely of AMG mode) results in a notable performance increase, with $AP_L^e$ improving from **17.3** to **20.3**. Furthermore, even with EMR and USR applied, incorporating MMG significantly improves performance, raising $AP_L^e$ from **40.8** to **43.6**. This emphasizes the crucial role of MMG in generating refined object-level masks, leading to enhanced ES performance. Fig. 6 also visually illustrates how MMG refines SAM's object-level masks, such as for the person on the right and the painting on the wall.

**Effectiveness of EMR Module.** The addition of the EMR module significantly improves performance over both the baseline and the MMG-only setup. When EMR is incorporated alongside MMG and USR, performance improves notably from **38.1** to **43.6** in $AP_L^e$, emphasizing EMR's effectiveness in refining overlapping masks and enhancing the accuracy of entity-level segmentation. Fig. 6 also demonstrates that, with the help of EMR, the identical person in the MMG output is effectively fused into a single entity mask.

**Effectiveness of USR Module.** Incorporating the USR module improves the performance by **5.4** $AP_L^e$ over the baseline. When adding USR into MMG and EMR modules separately, the performance improves by **17.8** and **11.3** in $AP_L^e$, respectively. Additionally, adding USR to the MMG and EMR combination further improves $AP_L^e$ from **35.0** to **43.6**, emphasizing USR's ability to refine under-segmented areas effectively. Fig. 6 also demonstrates the EMR's unsegmented regions and the USR's additional masks.

**Ablation study of Hyperparameters.** Tab. 4 presents the ablation study on hyperparameters and point prompts. The optimal values for each hyperparameter are as follows: $\theta_O$ achieves the best performance at 0.8, $\gamma_O$ at 0.6, and $\delta$ at 0.05. The optimal value for $\tau$ is 0.1, while $\rho$ is 0.1. For the number of point prompts, the best performance is obtained with 32/64, yielding an $AP^e$ of 43.6. *Further detailed analysis can be found in the Supplement.*

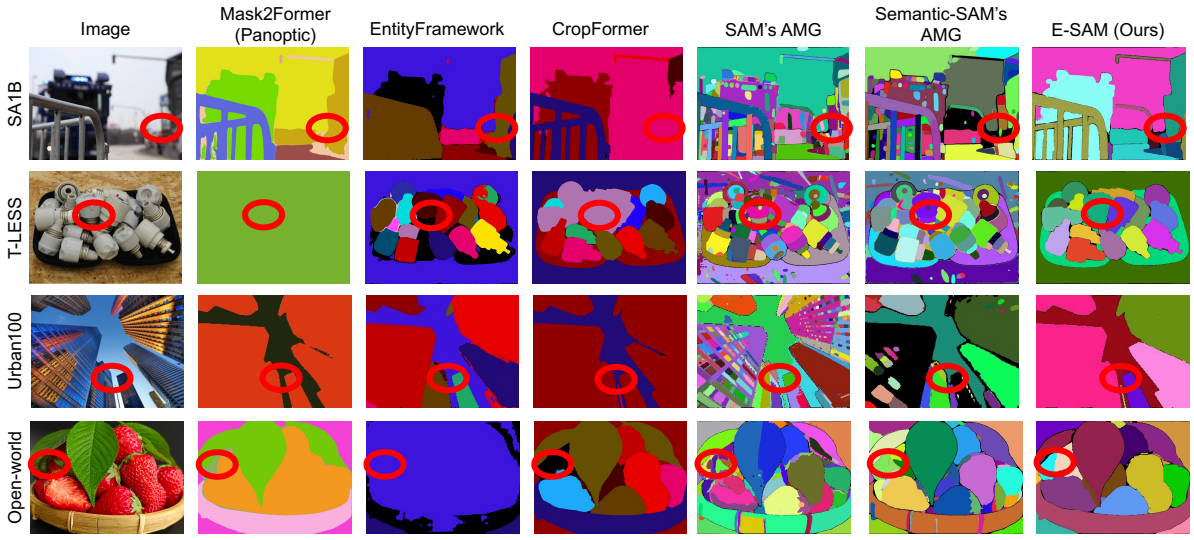


Figure 7. Example visualizations comparing various methods on SA1B [18], T-LESS [13], Urban100 [14] and open-world images.

| Method | Backbone | Params (M) | Inference (s) |
|---|---|---|---|
| ○ Mask2Former [4] | Swin-L | 234 | 0.26 |
| ○ EntityFramework [30] | FPN | - | 0.34 |
| ○ CropFormer [29] | Swin-L | 234 | 1.08 |
| ☆ E-SAM (Ours) | **ViT-H** | 636 | **9.84** |
| | SAM (Encoder) | 636 | 5.89 |
| | MMG | - | 1.22 |
| | EMR | - | 1.58 |
| | USR | - | 1.15 |

Table 5. Comparison of model complexity and inference time.

### 4.4. Other Analysis

**Performance of Object-Level Mask from SAM.** In this section, we discuss the performance of SAM's object-level masks (See Tab. 2). When SAM returns only the object-level mask map, the reduction in overlapping part- and subpart-level masks improves $AP_L^e$ scores from **17.0** to **23.7**. This demonstrates the feasibility of converting SAM's object-level masks into ES masks. Notably, our E-SAM consistently surpasses SAM$^O$ across all backbones, underscoring its novelty and effectiveness.

**Inference Time Comparision.** In this section, we discuss the inference time of E-SAM. As shown in Tab. 5, E-SAM takes **9.84** seconds on average to process a high-resolution image with the ViT-H backbone, which is higher than existing ES methods. However, this trade-off is justified by the avoidance of substantial training costs, with E-SAM introducing efficient post-processing operations. Notably, the SAM encoder accounts for **5.89** seconds, while each proposed module operates around **1** second. Given the strong generalization capabilities demonstrated in Fig. 7, the additional inference time is warranted.

**Superpixels Method Discussion.** Although our E-SAM leverages superpixels in all three modules, its novelty does not stem from a simple combination of superpixels and SAM. *Detailed comparisons and visualizations are provided in the Supplement.*

**Robustness Comparisons.** In this section, we test the robustness of E-SAM to address concerns about its performance being limited to the EntitySeg dataset. Fig. 7 demonstrates that E-SAM produces more robust ES results across other segmentation datasets and online open-world images. To further evaluate the real-world robustness of E-SAM, we tested it on 360 images (indoor/outdoor, real/synthetic). *More comparisons and failure case analyses can be found in the Supplement.*

## 5. Conclusion and Future Work

In this paper, we introduced a novel framework – E-SAM that enhances SAM's AMG for entity segmentation without additional training overhead. Our framework integrates MMG for refining multi-level mask generation, EMR for addressing overlapping segments and fusing entity-level masks, and USR for refining under-segmented regions. Extensive comparative experiments and ablation studies demonstrate that E-SAM achieves state-of-the-art performance and validates the effectiveness of each module.

**Future Work.** In future work, we plan to improve E-SAM's inference speed through parallel and multi-threaded processing in EMR and USR. We also aim to compress the ViT-H backbone or distill E-SAM into a compact student model to enhance deployment efficiency.

**Acknowledgement.** This work is supported by the MOE AcRF Tier 1 SSHR-TG Incubator Grant FY24 (Grant No. RSTG7/24), the National Natural Science Foundation of China (Grant No. 62206069, U22B2060), the National Key R&D Program of China (Grant No. 2023YFF0725100), and the Guangdong-Hong Kong Technology Innovation Joint Funding Scheme (Project No. 2024A0505040012).

## References


[1] Rui Cao, Chuanxin Song, Biqi Yang, Jiangliu Wang, Pheng-Ann Heng, and Yun-Hui Liu. Adapting segment anything model for unseen object instance segmentation. arXiv preprint arXiv:2409.15481, 2024. 2

[2] Shengcao Cao, Jiuxiang Gu, Jason Kuen, Hao Tan, Ruiyi Zhang, Handong Zhao, Ani Nenkova, Liang-Yan Gui, Tong Sun, and Yu-Xiong Wang. Sohes: Self-supervised open-world hierarchical entity segmentation. arXiv preprint arXiv:2404.12386, 2024. 2, 3

[3] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Wei Shen, Lingxi Xie, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, et al. Segment anything in 3d with nerfs. Advances in Neural Information Processing Systems, 36:25971–25990, 2023. 3

[4] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 1280–1289, 2021. 6, 8

[5] Yangming Cheng, Liulei Li, Yuanyou Xu, Xiaodi Li, Zongxin Yang, Wenguan Wang, and Yi Yang. Segment and track anything. arXiv preprint arXiv:2305.06558, 2023. 3

[6] Maximilian Durner, Wout Boerdijk, Martin Sundermeyer, Werner Friedl, Zoltán-Csaba Márton, and Rudolph Triebel. Unknown object segmentation from stereo images. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4823–4830. IEEE, 2021. 2

[7] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. International Journal of Computer Vision, 59:167–181, 2004. 4

[8] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. IEEE Transactions on Intelligent Transportation Systems, 22(3):1341–1360, 2020. 2

[9] Monica Gruosso, Nicola Capece, and Ugo Erra. Human segmentation in surveillance video with deep learning. Multimedia Tools and Applications, 80(1):1175–1199, 2021. 2

[10] Haoyu Guo, He Zhu, Sida Peng, Yuang Wang, Yujun Shen, Ruizhen Hu, and Xiaowei Zhou. Sam-guided graph cut for 3d instance segmentation. arXiv preprint arXiv:2312.08372, 2023. 3

[11] Arun Hampapur, Lisa Brown, Jonathan Connell, Sharat Pankanti, Andrew Senior, and Yingli Tian. Smart surveillance: applications, technologies and implications. In Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint, pages 1133–1138. IEEE, 2003. 2

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. 2017. 6

[13] Tomáš Hodan, Pavel Haluza, Štepán Obdržálek, Jiri Matas, Manolis Lourakis, and Xenophon Zabulis. T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. In 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 880–888. IEEE, 2017. 8

[14] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5197–5206, 2015. 8

[15] Yuhao Huang, Xin Yang, Lian Liu, Han Zhou, Ao Chang, Xinrui Zhou, Rusi Chen, Junxuan Yu, Jiongquan Chen, Chaoyu Chen, et al. Segment anything model for medical images? Medical Image Analysis, 92:103061, 2024. 3

[16] Lei Ke, Martin Danelljan, Xia Li, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Mask transfiner for high-quality instance segmentation. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 4402–4411, 2021. 6

[17] Mariia Khan, Yue Qiu, Yuren Cong, Bodo Rosenhahn, Jumana Abu-Khalaf, and David Suter. Segment any object model (saom): Real-to-simulation fine-tuning strategy for multi-class multi-instance segmentation. In 2024 IEEE International Conference on Image Processing (ICIP), pages 582–588. IEEE, 2024. 3

[18] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 4015–4026, 2023. 2, 3, 5, 6, 8

[19] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. International journal of computer vision, 128(7):1956–1981, 2020. 6

[20] Hyeokjun Kweon and Kuk-Jin Yoon. From sam to cams: Exploring segment anything model for weakly supervised semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 19499–19509, 2024. 3

[21] Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Jianwei Yang, Chunyuan Li, Lei Zhang, and Jianfeng Gao. Semantic-sam: Segment and recognize anything at any granularity. arXiv preprint arXiv:2307.04767, 2023. 1, 2, 6

[22] Jiaqi Li, Miaozeng Du, Chuanyi Zhang, Yongrui Chen, Nan Hu, Guilin Qi, Haiyun Jiang, Siyuan Cheng, and Bozhong Tian. Mike: A new benchmark for fine-grained multimodal entity knowledge editing. arXiv preprint arXiv:2402.14835, 2024. 2

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pages 740–755. Springer, 2014. 5, 6

[24] Songtao Liu, Di Huang, and Yunhong Wang. Adaptive nms: Refining pedestrian detection in a crowd. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 6459–6468, 2019. 4

[25] Jun Ma, Yuting He, Feifei Li, Lin Han, Chenyu You, and Bo Wang. Segment anything in medical images. Nature Communications, 15(1):654, 2024. 3

[26] D Man and A Vision. A computational investigation into the human representation and processing of visual information. WH San Francisco: Freeman and Company, San Francisco, 1:1, 1982. 1

[27] Maciej A Mazurowski, Haoyu Dong, Hanxue Gu, Jichen Yang, Nicholas Konz, and Yixin Zhang. Segment anything model for medical image analysis: an experimental study. Medical Image Analysis, 89:102918, 2023. 3

[28] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In 18th international conference on pattern recognition (ICPR'06), pages 850–855. IEEE, 2006. 2, 3

[29] Lu Qi, Jason Kuen, Weidong Guo, Tiancheng Shen, Jiuxiang Gu, Jiaya Jia, Zhe Lin, and Ming-Hsuan Yang. High-quality entity segmentation. arXiv preprint arXiv:2211.05776, 2022. 1, 2, 3, 5, 6, 8

[30] Lu Qi, Jason Kuen, Yi Wang, Jiuxiang Gu, Hengshuang Zhao, Philip Torr, Zhe Lin, and Jiaya Jia. Open world entity segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(7):8743–8756, 2022. 1, 2, 3, 6, 8

[31] Lu Qi, Lehan Yang, Weidong Guo, Yu Xu, Bo Du, Varun Jampani, and Ming-Hsuan Yang. Unigs: Unified representation for image generation and segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6305–6315, 2024. 2, 3

[32] Frano Rajič, Lei Ke, Yu-Wing Tai, Chi-Keung Tang, Martin Danelljan, and Fisher Yu. Segment anything meets point tracking. arXiv preprint arXiv:2307.01197, 2023. 3

[33] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. Advances in neural information processing systems, 35:25278–25294, 2022. 6

[34] Han Shu, Wenshuo Li, Yehui Tang, Yiman Zhang, Yihao Chen, Houqiang Li, Yunhe Wang, and Xinghao Chen. Tinysam: Pushing the envelope for efficient segment anything model. arXiv preprint arXiv:2312.13789, 2023. 2, 3

[35] Jianan Wang, Guansong Lu, Hang Xu, Zhenguo Li, Chunjing Xu, and Yanwei Fu. Manitrans: Entity-level text-guided image manipulation via token-wise semantic alignment and generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10707–10717, 2022. 2

[36] XuDong Wang, Jingfeng Yang, and Trevor Darrell. Segment anything without supervision. arXiv preprint arXiv:2406.20081, 2024. 2, 3

[37] Yikai Wang, Jianan Wang, Guansong Lu, Hang Xu, Zhenguo Li, Wei Zhang, and Yanwei Fu. Entity-level text-guided image manipulation. arXiv preprint arXiv:2302.11383, 2023. 2

[38] Qi Wen, Jirui Yang, Xue Yang, and Kewei Liang. Patchdct: Patch refinement for high quality instance segmentation. ArXiv, abs/2302.02693, 2023. 6

[39] Christopher Xie, Yu Xiang, Arsalan Mousavian, and Dieter Fox. Unseen object instance segmentation for robotic environments. IEEE Transactions on Robotics, 37(5):1343–1359, 2021. 2

[40] Yinsong Xu, Jiaqi Tang, Aidong Men, and Qingchao Chen. Eviprompt: A training-free evidential prompt generation method for segment anything model in medical images. arXiv preprint arXiv:2311.06400, 2023. 2, 3

[41] Jinyu Yang, Mingqi Gao, Zhe Li, Shang Gao, Fangjing Wang, and Feng Zheng. Track anything: Segment anything meets videos. arXiv preprint arXiv:2304.11968, 2023. 3

[42] Yunhan Yang, Xiaoyang Wu, Tong He, Hengshuang Zhao, and Xihui Liu. Sam3d: Segment anything in 3d scenes. arXiv preprint arXiv:2306.03908, 2023. 3

[43] Yingda Yin, Yuzheng Liu, Yang Xiao, Daniel Cohen-Or, Jingwei Huang, and Baoquan Chen. Sai3d: Segment any instance in 3d scenes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3292–3302, 2024. 3

[44] Haiyue Yuan, Ali Raza, Nikolay Matyunin, Jibesh Patra, and Shujun Li. A graph-based model for vehicle-centric data sharing ecosystem. arXiv preprint arXiv:2410.22897, 2024. 2

[45] Kaidong Zhang and Dong Liu. Customized segment anything model for medical image segmentation. arXiv preprint arXiv:2304.13785, 2023. 3

[46] Jiawen Zhu, Zhenyu Chen, Zeqi Hao, Shijie Chang, Lu Zhang, Dong Wang, Huchuan Lu, Bin Luo, Jun-Yan He, Jin-Peng Lan, et al. Tracking anything in high quality. arXiv preprint arXiv:2307.13974, 2023. 3