# E-SAM: Training-Free Segment Every Entity Model
# –Supplementary Material–

## Abstract

*Due to the lack of space in the main paper, we provide more details of the motivation, proposed methods, and experiment results in the supplementary material. Sec. 1 elaborates on the motivation behind our E-SAM, while Sec. 2 provides detailed insights into the design of our training-free framework. Sec. 3 presents extensive comparisons of visual results between our E-SAM and prior SOTA methods, along with further discussions.*

## 1. More Details of Motivation

**Motivation of Optimizing SAM for Entity Segmentation task.** In this section, we provide a more detailed explanation of our motivation behind this work. Inspired by [17, 18], Entity Segmentation (ES) is an advanced computer vision task focused on segmenting all perceptually distinct entities in an image, without relying on predefined class labels. Unlike traditional segmentation tasks that require a fixed taxonomy of categories, ES aims to partition an image into its meaningful components, treating each distinct object, part, or even background entity as a separate segment. This makes ES particularly suitable for open-world scenarios where predefined categories are impractical or unavailable. The task inherently aligns with the human visual system, which intuitively segments a scene into entities based on perceptual cues such as shape, color, and spatial context. This perceptual alignment has garnered significant attention, as it enables a more flexible and detailed understanding of visual data, which can be applied to various real-world applications. Compared to existing methods specifically designed for the ES task, we find that training these models demands substantial annotated data and considerable computational resources, posing a significant barrier for researchers with limited resources. For instance, the ES benchmark dataset, EntitySeg [17], contains numerous high-resolution images, which increases both the complexity and computational demands of the training process.

Segment Anything Model (SAM) [11] is a powerful foundational model designed to address a wide range of segmentation tasks. It employs a novel architecture that integrates efficient mask prediction with an interactive prompting system, allowing SAM to achieve remarkable flexibility in segmenting different types of regions with minimal user guidance. SAM's unique zero-shot capability makes it suitable for generalization across various segmentation scenarios without the need for task-specific fine-tuning. Its versatility has led to successful applications in multiple segmentation domains, including instance segmentation [10, 22], semantic segmentation [13, 24], and panoptic segmentation [16, 21] tasks. However, despite its broad applicability, SAM has not yet been explored for Entity Segmentation (ES) tasks. ***In this work, we are the first to explore and effectively adapt SAM to improve its performance specifically for ES.***

Notably, SAM's Automatic Mask Generation (AMG) mode is designed to autonomously generate segmentation masks for different instances within an image. It works by using uniformly sampled point prompts across the entire image to predict multiple masks, each representing a different segment or object within the image. AMG's goal is to "segment everything" by efficiently covering all distinguishable objects, regardless of their semantic class or characteristics, making it suitable for a wide range of segmentation tasks without relying on explicit annotations or training data. Ideally, both SAM's AMG and ES tasks share a common objective: to distinguish and delineate all perceptually separate entities present in an image. Therefore, We propose to follow the design mechanism of SAM's AMG. in a training-free manner, without modifying any internal parameters of SAM itself. Our objective is to refine SAM's multi-granularity masks generated from uniformly sampled points into entity-level masks, thereby effectively mitigating the over-segmentation and under-segmentation issues without requiring additional training or fine-tuning.

**Motivation of Using Superpixels.** In this section, we provide further explanation of why we utilize superpixels in our approach. Intuitively, as shown in Fig. 1, SAM's AMG mode uses a naive NMS method, which results in some areas having excessive overlapping masks while others are left without adequate mask coverage. This demonstrates the over-segmentation and under-segmentation issues we mentioned in the main paper regarding SAM's AMG. In con-

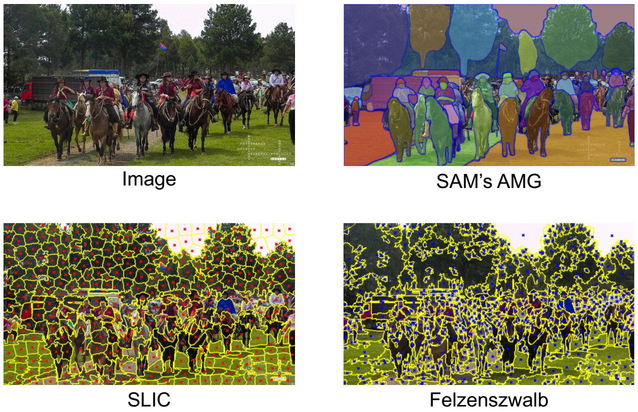

Figure 1. Visual Comparison of Superpixel Methods: SLIC [1], Felzenszwalb [6], and AMG's result

trast, superpixels ensure that every pixel in the image is clustered into a segment based on its color, texture, or appearance. By leveraging superpixels, our E-SAM can effectively address overlapping masks while also recognizing the clustering relationships within under-segmented areas, leading to more comprehensive entity segmentation. It can be observed that superpixels are employed in the design of all three modules: Multi-level Mask Generation (MMG), Entity-level Mask Refinement (EMR), and Under-Segmentation Refinement (USR). This highlights our effective and extensive usage of superpixels, leveraging their capabilities across the entire framework to enhance segmentation performance. While using lightweight clustering or segmentation networks could potentially yield faster or higher-quality clustering results and help E-SAM achieve better segmentation, it would inevitably reduce the novelty of our approach by showing reliance on external clustering outputs. To avoid this, we chose to employ the most commonly used superpixel clustering methods in our approach, focusing instead on the effectiveness of the design of our three modules.

For the superpixel generation, our E-SAM considers two of the most common methods: SLIC [1] and Felzenszwalb [6]. Fig. 1 illustrates the differences in the visual results of the two methods, where yellow lines indicate superpixel boundaries and red/blue points represent the centroids of each superpixel. Intuitively, SLIC generates uniformly sized and distributed superpixels, often causing multiple objects to be contained within a single superpixel. In contrast, Felzenszwalb adapts to object density, creating smaller superpixels in densely populated areas and larger ones in sparse regions. Given our emphasis on superpixel density, E-SAM incorporates Felzenszwalb to produce superpixels that better align with the distribution of entities in the image.

**Algorithm 1:** Overlap Mask Fusion

```
Input: Object-level masks $\hat{M}_O^{32}$ after MMG. $M_O^{64}$
       and $M_B^{64}$ from mask gallery $G$.
Extract overlap mask pairs in $\hat{M}_O^{32}$.
while Overlap pairs exist do
    Initiate an array $v$ to record whether a mask of
      $\hat{M}_O^{32}$ is visited.
    for Each overlap mask pair $M_p^{32}, M_q^{32}$ do
        if $v[M_p^{32}]$ or $v[M_q^{32}]$ then
            continue
        $OR_p^q = M_p^{32} \cap M_q^{32}$
        $U_p^q = M_p^{32} \cup M_q^{32}$
        if $OR_p^q$ occupies less than $\tau_l$ of larger mask
          in $M_p^{32}, M_q^{32}$ then
            Cut $OR_p^q$ from the larger mask and
              merge it into a smaller one.
            $v[M_p^{32}], v[M_q^{32}] = True, True$
            continue
        Initiate a boolean variable $m$ $False$.
        if $OR_p^q$ occupies larger than $\tau_s$ of smaller
          mask in $M_p^{32}, M_q^{32}$ then
            if Corresponding prompt of $\hat{M}_B^{64}$ lying
              in $OR_p^q$ exists then
                $M_{B,p,q}^{64}$ is the mask of those prompts.
                  Filter $M_{B,p,q}^{64}$ by $IoU$ larger than $\varphi$
                  with $U_p^q$ to obtain $F_{B,p,q}^{64}$.
                $m = True$
                if $F_{B,p,q}^{64}$ is not empty then
                    Use $U_p^q$ as the merge result of
                      $M_p^{32}, M_q^{32}$ to update $\hat{M}_O^{32}$.
                    $v[M_p^{32}], v[M_q^{32}] = True, True$
                else
                    Use the area proportion of
                      $M_{B,p,q}^{64}$ in $OR_p^q$ $M_p^{32}$ and $M_q^{32}$
                      to vote for the belonging of
                      $OR_p^q$.
                    Update $\hat{M}_O^{32}$.
                    $v[M_p^{32}], v[M_q^{32}] = True, True$
        if not $m$ then
            Cut $OR_p^q$ from the mask with smaller
              $pred\_iou\ of\ SAM$ and merge it into
              the other one.
              $v[M_p^{32}], v[M_q^{32}] = True, True$
    Extract overlap mask pairs in $\hat{M}_O^{32}$.
```

## 2. More Details of Methodology

Due to space limitations in the main paper, we provide additional explanations of the novel design within the E-SAM framework using accompanying pseudocode. Sec. 2.1 provides further details about EMR, while Sec. 2.2 elaborates on the design aspects of USR.

### 2.1. More detailed in EMR module

In this section, we describe our EMR process in detail, which aims to enhance the object-level masks generated by the MMG module into accurate entity-level masks. The EMR process consists of several key steps, each contributing to the overall refinement and improvement of segmentation quality. Below, we provide a detailed breakdown of the EMR framework.

**Mask Overlap Extraction** The EMR process begins by extracting overlapping mask pairs from the object-level masks, denoted as $\hat{M}_O^{32}$, which are obtained after applying MMG. Specifically, as shown in Alg. 1, we identify all the mask pairs that have overlapping regions, which are stored and analyzed in subsequent steps. These overlap pairs are crucial for understanding the extent of redundancy in mask coverage and serve as the foundation for refinement. Once the overlapping mask pairs $(\hat{M}_p^{32}, \hat{M}_q^{32})$ are identified, we evaluate the overlapping region $OR_p^q$. If the area of overlap is relatively small, below a threshold $\tau_1$ in comparison to the larger mask, the overlapping section is simply cut, and the smaller mask is merged into the larger one. If the overlapping area is significant, we then consider prompts from the mask gallery, $G$, to determine if further adjustments are required. The EMR module leverages additional mask information from a mask gallery generated by 64-point per-side sampling, denoted as $P^{64}$. The mask gallery includes both object-level masks $M_O^{64}$ and best-level masks $M_B^{64}$, which provide additional guidance for refining overlaps. Based on these prompts, overlapping masks $\hat{M}_p^{32}$ and $\hat{M}_q^{32}$ are filtered using an Intersection over Union (IoU) criterion to obtain a subset of masks, $F_{B,p,q}^{64}$, that are suitable candidates for merging. The mask with the highest confidence score is selected as the merge candidate, leading to a refined and more coherent mask representation.

**Adjacent Mask Fusion.** Next, we construct a similarity matrix, $S_C$, to evaluate the relationship between superpixel centroids within the mask map $M_S$. This matrix helps to determine which masks are adjacent and whether they should be merged based on similarity. Using the top-k most similar centroids for each mask, we establish the adjacent mask similarity matrix, $S_M$, which helps to identify which adjacent masks share similar features. This similarity-based merging effectively reduces redundancy and refines the entity-level representation. As shown in Alg. 2, once the similarity evaluations and merging operations are completed, the refined masks $\hat{M}_O^{32}$ are updated to remove overlap and ensure consistency. The process iterates until all overlapping masks are processed and no further overlaps exist. The output of the EMR module is an entity-level map $M_E$, which provides refined and high-quality segmentation suitable for entity segmentation tasks.

**Algorithm 2:** Adajacent Mask Fusion

```
Input: Object-level masks $\hat{M}_O^{32}$ after MMG. $M_B^{64}$
       from mask gallery $G$. Superpixel centroids
       $C$. Image feature $F$ from SAM encoder.
Use normalized feature cosine similarity to
  construct centroid similarity mask $S_C$.
Initiate a boolean variable $m$ $True$.
while $m$ do
    $m = False$
    Initiate a Boolean array $V$ as $False$ to record
      whether a mask of $\hat{M}_O^{32}$ is visited.
    Initiate an all-minus-one square matrix $S_M$ with
      the side length the same as the number of
      masks in $\hat{M}_O^{32}$.
    for Each mask pair $\hat{M}_i, \hat{M}_j$ in $\hat{M}_O^{32}$ do
        $C_i$ is the set of all superpixels in mask $\hat{M}_i$.
          $C_j$ is the set of all superpixels in mask $\hat{M}_j$.
          $S_M(\hat{M}_{c_i}, \hat{M}_{c_j}) =$
          $\frac{1}{|C_i|}\sum_{c_i \in C_i} |\{c_j \mid c_j \in C_j \wedge c_j \in \text{Top}_k(S_C(c_i, \cdot))\}|$
    Choose top-k similar mates for each mask and
      update $S_M$.
    for Each mask pair $\hat{M}_i, \hat{M}_j$ in $S_M$ do
        if $V[\hat{M}_i]$ or $V[\hat{M}_j]$ then
            continue
        $U_i^j = \hat{M}_i \cup \hat{M}_j$
        if $U_i^j$ is a part of any mask in $\hat{M}_O^{32}$ then
            Update $\hat{M}_i$ and $\hat{M}_j$ by $U_i^j$.
            $V[\hat{M}_i], V[\hat{M}_j] = True, True$
            $m = True$
```

### 2.2. More detailed in USR module

The USR module (as shown in Alg. 3) addresses the under-segmentation issues in SAM's AMG-generated outputs, ensuring comprehensive coverage of all distinct entities. Below, we provide an overview of the USR framework: The USR process starts by identifying areas not covered in the entity-level map $M_E$, which is generated by EMR. These regions are represented as $N_{M_E}$, indicating parts of the image that require additional segmentation masks. To address under-segmented areas, USR generates additional point prompts based on superpixel information. Specifically, for each superpixel $M_S^i$ that is not covered by $M_E$, the centroid is computed. If any part of the superpixel is covered by part-level or subpart-level masks ($M_P^{32}$ or $M_{SP}^{32}$), the centroid of the corresponding mask is used as the additional prompt. Otherwise, the centroid of the superpixel itself is used as the prompt. These prompts are compiled into an array $P_A$ and used to generate additional masks, denoted as $M_A^k$, through SAM's prompt encoder and decoder. Once the additional masks $M_A^k$ are generated, the USR module evaluates them against the current entity-level map $M_E$. For each mask $M_A^k$, an Intersection-over-Union (IoU) score is computed with $M_E$. If the IoU exceeds a specified threshold $\rho$, the mask is merged with the entity-level map $M_E$. If the IoU is below $\rho$, the mask is retained as an independent entity-level mask. This evaluation ensures that only relevant masks are incorporated to refine segmentation, minimizing redundancy. To avoid generating an excessive number of overlapping part-level masks, a greedy algorithm is employed. The goal is to use a minimal number of masks from $M_A^k$ to fill in the under-segmented regions of $M_E$, optimizing segmentation quality and maintaining efficiency. This refinement strategy allows USR to significantly enhance the completeness and robustness of the segmentation, particularly in regions with complex boundaries or densely packed entities. After iterating through the additional prompts and refining the masks, the USR module updates the entity-level map $M_E$, producing a more refined and comprehensive entity segmentation map. This output ensures that all perceptually distinct entities are effectively segmented, addressing the under-segmentation limitations of SAM's AMG mode.

**Algorithm 3:** Under-Segmentation Refnement

```
Input: Entity-level map $M_E$. Part-level masks $M_P^{32}$
       and subpart-level masks $M_{SP}^{32}$. Superpixel
       map $M_S$. SAM prompt encoder and decoder
       $SED$. Image feature $F$ from SAM encoder.
Find the unsegmented area $N_{M_E}$ in the image by $M_E$.
Initiate empty coordinate array $P_A$ to record additional prompts.
for Each superpixel $M_{S_i}$ in $M_S$ do
    if Any pixel of $M_{S_i}$ is not segmented by $M_E$ then
        $M_{P,S_i}$ is a mask in $M_P^{32}$ or $M_{SP}^{32}$ that covers $M_{S_i}$.
        if $M_{P,S_i}$ exists then
            Append the centroid of $M_{P,S_i}$ to $P_A$.
        else
            Append the centroid of $M_{S_i}$ to $P_A$.
$M_{P_A} \equiv SED(F, P_A)$
for Each superpixel $M_i$ in $M_S$ do
    if Less than $\rho_u$ of $M_i$ is in $N_{M_E}$ then
        Drop $M_i$.
    if More than $\rho_s$ of $M_i$ is covered by one mask in $M_E$ then
        Merge $M_i$ with the mask.
        Drop $M_i$.
Use a greedy algorithm to choose masks in $M_S$ to
  fill $N_{M_E}$ and update $M_E$ to $\check{M}_E$.
```

# 3. More Details of Experiments

## 3.1. More Details of Datasets

For qualitative comparison, we adopt the validation set of the EntitySeg dataset [17]. The EntitySeg dataset comprises 33,227 high-resolution images aggregated from COCO [15], ADE20K[25], Pascal VOC[5], LAION [19], Open Images [12], and many other famous datasets. It features precise, class-agnostic entity masks for open-world segmentation. Unlike prior datasets, EntitySeg emphasizes high-quality annotations and complex scenarios. As 86.23% of original images are of high resolution (over 1000px×1000px), a low-resolution version is created by resizing images to below 800px×1333px. The validation set of EntitySeg has 1,314 images in total.

To test the robustness of the proposed method, we conduct visual comparison in the paper and the supplementary material with images from SA1B [11], T-LESS [8], Urban100 [9], NDD20 [20], LVIS [7], V2X-SIM [14], Sketch [4], and other self-collected images from the internet.

## 3.2. More Details of Implementation

In this work, we focus on optimizing the performance of SAM's AMG mode, which led us to directly adopt the pretrained SAM backbones: ViT-B, ViT-L, and ViT-H. Additionally, we select the top $k = 10$ most similar centroids to centroid $c$ for constructing the adjacent mask similarity matrix. For evaluation, we adopt different strategies depending on the dataset. For the benchmark dataset, EntitySeg, we utilize the benchmark evaluation metric $AP^e$, which is specifically designed for entity segmentation. $AP^e$ is computed by averaging precision over all entities at different IoU thresholds, considering class-agnostic mask predictions, with rankings based on confidence scores. This metric measures the quality of segmentation across multiple thresholds, with a strong emphasis on minimizing the presence of overlapping masks, ensuring that each entity is distinctly represented. The usage of $AP^e$ helps evaluate the effectiveness of our approach in producing clean, non-overlapping masks that are crucial for accurate entity-level segmentation. For other public datasets [7, 20], we visualize

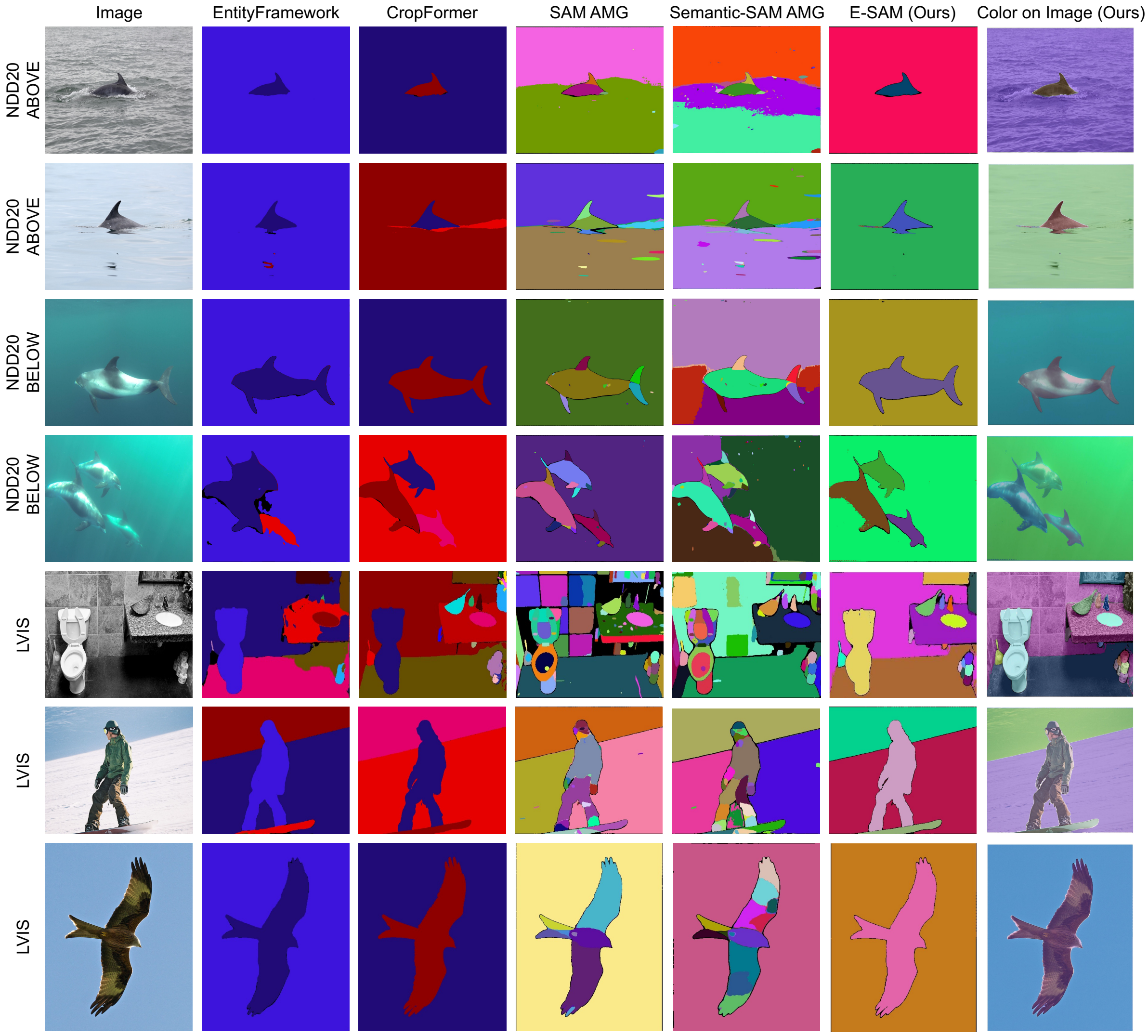

Figure 2. Example visualizations comparing various methods on NDD20 (above and below water) [20] and LVIS (multiple objects and single one) [7].

| Method | $\mathbf{AP}^e$ | $\mathbf{AP}^e_{50}$ | $\mathbf{AP}^e_{75}$ |
|---|---|---|---|
| CropFormer | 18.16 | 28.99 | 19.65 |
| Mask2Former | 17.88 | **30.42** | 18.33 |
| **E-SAM (Ours)** | **19.21** | 29.55 | **20.32** |

Table 1. Comparisons on panoptic segmentation performance on COCO val2017 dataset [3].

the segmentation results to provide qualitative comparisons with existing state-of-the-art methods and further validate the robustness of our E-SAM framework.

### 3.3. Extended Experiments

**Comparisons on Panoptic Dataset.** Extensive experiments were conducted to compare E-SAM against CropFormer [17] and Mask2Former (panoptic) [3] using 100 images sampled from the COCO val2017 dataset [3]. As demonstrated in Tab. 1, E-SAM achieves superior overall panoptic segmentation performance, surpassing both methods by a margin of at least 1.05% $AP^e$. However, in terms of $\text{AP}^e_{50}$, Mask2Former outperforms E-SAM by 0.87% $\text{AP}^e_{50}$. This discrepancy can be attributed to the IoU threshold of 0.5 employed by $\text{AP}^e_{50}$ for determining true positives, where Mask2Former, specifically designed for panoptic segmentation tasks, demonstrates better performance under this metric.

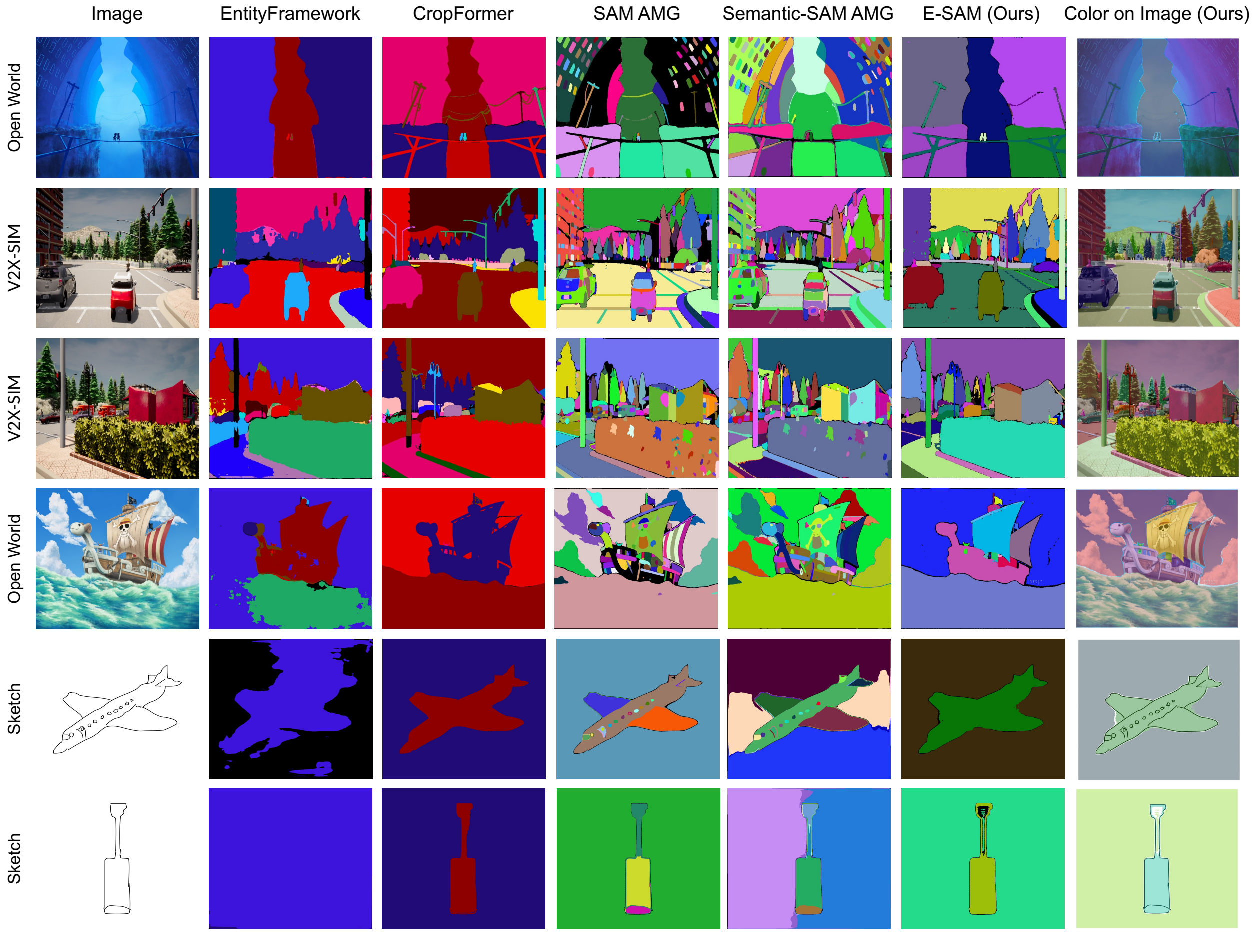

Figure 3. Qualitative comparison on open-world images (cartoon style), V2X-SIM (synthetic)[14], and Sketch[4].

### 3.4. Extended Visual Comparisons for 2D Images

Fig.2 shows the qualitative results of our methods and others to compare. Images are chosen from NDD20 [20] and LVIS [7] to see E-SAM's performance under a range of domains like near the water, below the water, multiple objects, and a single object within the monotone background. Obviously, E-SAM's result is better than other methods when segmenting pictures near the water, which is affected by the reflection, not water waves. For complex backgrounds with multiple objects (the third to last image), E-SAM shows great boundary detection ability inherited from SAM without over-segmentation. Such ability outstands when handling small objects clustered together. To demonstrate the robustness of the proposed method, we show more visualization examples from non-naturalistic images.

**Comparisons on Cartoons.** For the first two rows in Fig. 3, we conducted a visual comparison between prior SOTA entity segmentation methods and SAM-based methods using open-world cartoon-style images. It can be observed that our E-SAM outperforms all methods except CropFormer. Given that our approach is training-free, achieving comparable performance to CropFormer is an acceptable trade-off, demonstrating the practicality and efficiency of our method.

**Comparisons on Simulation Images.** As shown in the third and fourth rows of Fig. 3, E-SAM outperforms other methods, particularly when handling numerous similar objects in the background. This effectively demonstrates the robustness and effectiveness of the E-SAM framework. Notably, unlike CropFormer, which struggles with distinguishing same-class objects in the background and tends to exaggerate foreground elements, E-SAM achieves a better balance between foreground and background segmentation. For instance, in the third row, E-SAM accurately segments the trees in the background.

**Comparisons on Sketches.** E-SAM's performance is close to CropFormer when handling sketches. Since sketches consist of only lines, it confuses SAM and affects the information for E-SAM. In contrast to SAM and Semantic-SAM, our E-SAM effectively removes overlapping masks and successfully merges masks corresponding to the same entity. Given its training-free nature, this highlights the nov-

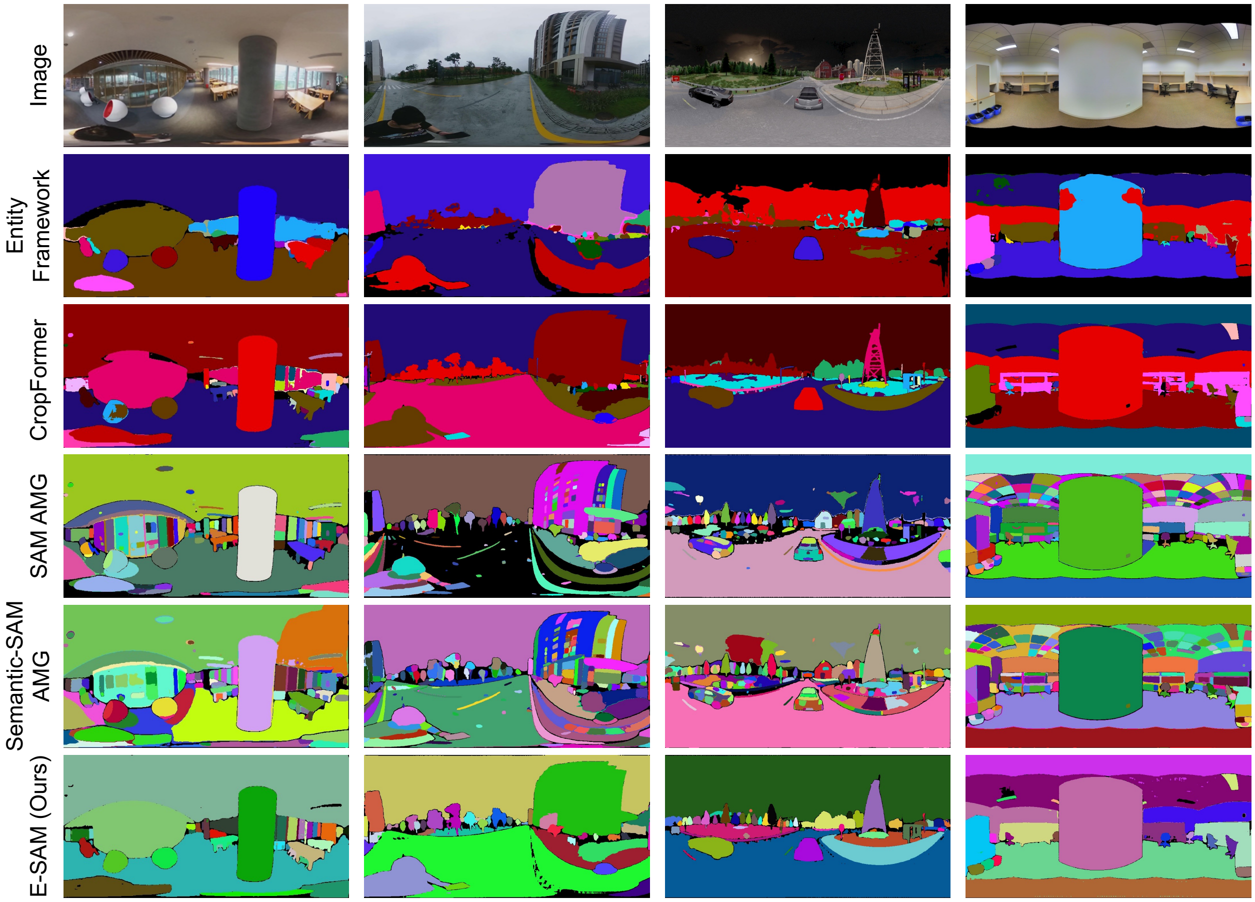

Figure 4. Segmentation visualization of various approaches for the 360 images in indoor&outdoor environment (self-collected), simulation [23], and Stanford2D3D [2] dataset.

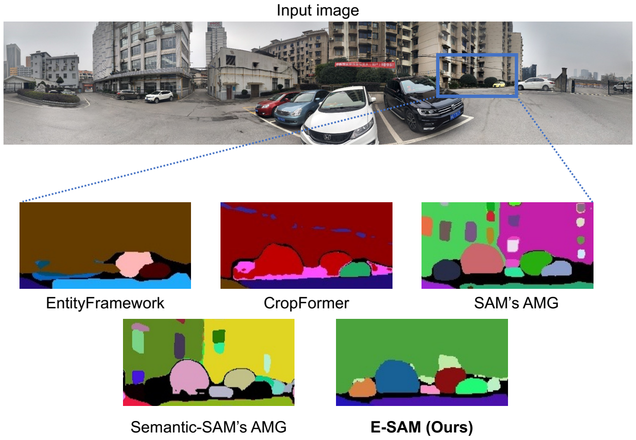

Figure 5. Sample detailed visualization of various approaches for the 360 image.

elty of our E-SAM even more clearly.

| $\theta_O$ | | $\gamma_O$ | | $\delta$ | |
|---|---|---|---|---|---|
| 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 2. Ablation study on hyperparameters and the number of point prompts. All performance is evaluated in each module.

### 3.5. Extended Visual Comparisons for 360 Images

Fig.4 and Fig.5 demonstrate segmentation results of various methods on 360° images. We chose to evaluate 360° images to validate the robustness of our E-SAM, specifically under challenging conditions of large FoV and severe distortion, to assess whether E-SAM maintains satisfactory performance. This exploration also considers the scalability

| Backbone | $\mathrm{AP}^e$ | $\mathrm{AP}^e_{50}$ | $\mathrm{AP}^e_{75}$ |
|---|---|---|---|
| ViT-B (Superpixel) | 5.22 | 10.70 | 4.24 |
| ViT-L (Superpixel) | 6.52 | 13.49 | 5.52 |
| ViT-H (Superpixel) | 6.66 | 13.71 | 5.56 |
| ViT-H (E-SAM) | **50.2** | **66.8** | **49.9** |

Table 3. E-SAM vs. Superpixel-based fusion.

of E-SAM for applications in 360-degree devices like VR, AR, or autonomous driving.

In Fig. 4, we compare our E-SAM with prior methods across indoor and outdoor open-world scenes, synthetic environments, and the indoor benchmark dataset Stanford2D3D [2]. In both indoor and outdoor open-world scenarios, E-SAM achieves accurate segmentation, such as the table in the first column, and demonstrates strong performance even for small objects near the equator of the image, significantly outperforming the other methods. In synthetic 360-degree environments, E-SAM maintains refined and detailed entity-level segmentation results, showing resilience against real-synthetic domain gaps. In the Stanford2D3D dataset, E-SAM consistently delivers precise masks for distant or smaller objects, while our three modules collaboratively reduce overlapping masks and effectively refine SAM AMG results into entity-level segmentation maps, outperforming SAM and Semantic-SAM.

In Fig. 5, we further zoom in on the details of our E-SAM and prior methods for 360 images to demonstrate the robustness of our approach. It is evident that, compared to existing ES methods, E-SAM accurately segments smaller objects and provides results that align more closely with human visual perception, such as distinguishing two bushes as separate entities, thereby reducing ambiguous segmentation. Compared to SAM and Semantic-SAM, our E-SAM effectively produces entity-level mask results for distorted and very small objects, even without additional training. This further highlights the practicality and effectiveness of our approach.

### 3.6. More Analysis

**Hyperparameters Discussion.** Our hyperparameter ablation study, presented in Table 2, systematically evaluates key architecture parameters. Through extensive experiments on the EntitySeg-LR dataset, we carefully calibrated $\theta_O$ and $\gamma_O$ using the MMG module, while $\delta$ and $\tau$ were optimized via MMG and EMR. The final tuning of $\rho$ and point-prompts-per-side was conducted with the complete E-SAM framework, achieving optimal performance and demonstrating superior effectiveness in our experimental settings.

**Comparison of Combine Superpixels with SAM.** Given that all three modules utilize superpixels for different purposes, concerns may arise that E-SAM's performance is overly dependent on the superpixel map. To address this, we conducted experiments by directly combining SAM's AMG with superpixels to generate the ES map (see Tab. 3). Specifically, when we fused the masks returned by SAM using superpixel guidance, we found that this direct integration actually degraded SAM's AMG performance. In contrast, our E-SAM demonstrates significant performance gains, *underscoring that its novelty and effectiveness stem from the design of the three modules rather than merely from combining superpixels with SAM's AMG*.

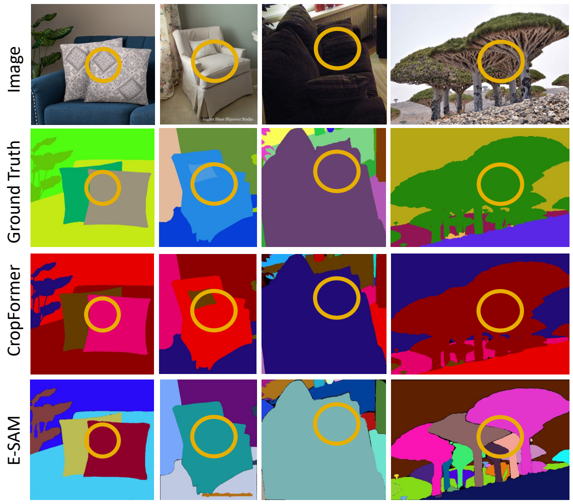

Figure 6. Example visualizations of failure cases.

**Limitation of E-SAM.** Although our current E-SAM demonstrates promising generalization capabilities, it still has several limitations. For instance, E-SAM retains SAM's computationally intensive image encoder, and the post-processing applied by our modules to SAM's outputs results in inference times that are significantly longer than those of ES-based methods. Moreover, Fig. 6 lists some failure cases. They may fall into two categories: (1) Since E-SAM inherits SAM's instance-level awareness, it lacks sensitivity to stuff entities ('tree'), leading to less satisfactory performance on EntitySeg. (2) Without ES dataset training, E-SAM cannot account for annotation biases such as backrest present in EntitySeg.

## References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. IEEE transactions on pattern analysis and machine intelligence, 34(11):2274–2282, 2012. 2

[2] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. arXiv preprint arXiv:1702.01105, 2017. 7, 8

[3] 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. 5

[4] Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? ACM Trans. Graph. (Proc. SIGGRAPH), 31(4):44:1–44:10, 2012. 4, 6

[5] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. International journal of computer vision, 88:303–338, 2010. 4

[6] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. International Journal of Computer Vision, 59:167–181, 2004. 2

[7] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 5356–5364, 2019. 4, 5, 6

[8] 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. 4

[9] 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. 4

[10] 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. 1

[11] 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. 1, 4

[12] 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. 4

[13] 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

[14] Yiming Li, Dekun Ma, Ziyan An, Zixun Wang, Yiqi Zhong, Siheng Chen, and Chen Feng. V2x-sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving. IEEE Robotics and Automation Letters, 7(4):10914–10921, 2022. 4, 6

[15] 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. 4

[16] Khoa Dang Nguyen, Thanh-Hai Phung, and Hoang-Giang Cao. A sam-based solution for hierarchical panoptic segmentation of crops and weeds competition. arXiv preprint arXiv:2309.13578, 2023. 1

[17] 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, 4, 5

[18] 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

[19] 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. 4

[20] Cameron Trotter, Georgia Atkinson, Matt Sharpe, Kirsten Richardson, A Stephen McGough, Nick Wright, Ben Burville, and Per Berggren. Ndd20: A large-scale few-shot dolphin dataset for coarse and fine-grained categorisation. arXiv preprint arXiv:2005.13359, 2020. 4, 5, 6

[21] Vibashan VS, Shubhankar Borse, Hyojin Park, Debasmit Das, Vishal Patel, Munawar Hayat, and Fatih Porikli. Possam: Panoptic open-vocabulary segment anything. arXiv preprint arXiv:2403.09620, 2024. 1

[22] Yunhan Yang, Xiaoyang Wu, Tong He, Hengshuang Zhao, and Xihui Liu. Sam3d: Segment anything in 3d scenes. arXiv preprint arXiv:2306.03908, 2023. 1

[23] Jiaming Zhang, Kailun Yang, Hao Shi, Simon Reiß, Kunyu Peng, Chaoxiang Ma, Haodong Fu, Philip HS Torr, Kaiwei Wang, and Rainer Stiefelhagen. Behind every domain there is a shift: Adapting distortion-aware vision transformers for panoramic semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024. 7

[24] Weiming Zhang, Yexin Liu, Xu Zheng, and Lin Wang. Goodsam: Bridging domain and capacity gaps via segment anything model for distortion-aware panoramic semantic segmentation. arXiv preprint arXiv:2403.16370, 2024. 1

[25] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 633–641, 2017. 4