BOUNDARY ADAPTATION FOR SPARSE SEGMENTATION REFINEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

To be computationally efficient, segmentation networks typically predict masks at a low resolution. These coarse masks have errors concentrated along the boundary, which are particularly significant for small objects. Refinement methods have been proposed to correct these coarse masks, often updating a sparse subset of pixels for computational efficiency. Existing sparse methods use entropy for sparse pixel selection, but in high-quality segmentation networks (like SAM) high-entropy pixels do not strongly correlate with pixels that require correction. We present a novel sparse pixel selection strategy for segmentation refinement called Boundary Adaptation, and a corresponding pointwise refinement architecture called Boundary-SAM. We show the proposed sparse refinement method outperforms high-quality mask-based methods on natural image datasets, a remote sensing dataset, and a CT scan dataset.

1 INTRODUCTION

Image segmentation is a core computer vision task, with applications such as autonomous perception (Zhao et al., 2023; Sun et al., 2020), image editing (Xu et al., 2017), remote sensing (Rahnemoonfar et al., 2021; Garnot & Landrieu, 2021; Xia et al., 2018), and assisted medical diagnoses (Antonelli et al., 2022; Bilic et al., 2023; Heller et al., 2021). To be computationally efficient, segmentation networks typically predict masks at a low resolution (Kirillov et al., 2023; Wang et al., 2025; Sun et al., 2019; He et al., 2017; Ke et al., 2022; Li et al., 2023; Kirillov et al., 2020; Ronneberger et al., 2015; Wang et al., 2020; Tian et al., 2020). These masks are then upsampled with bilinear interpolation, which fundamentally limits their segmentation quality. Consequently, segmentation errors of these coarse masks are concentrated along the boundary (Yuan et al., 2020; Tang et al., 2021). For small objects, where the boundary makes up a large fraction of the mask, these errors are particularly significant.

Segmentation refinement methods (or simply *refinement methods*) are designed to correct these coarse masks. They either refine the entire mask (Lin et al., 2025; Shen et al., 2022; Cheng et al., 2020), or a sparse set of pixels (Liu et al., 2024; Tang et al., 2021; Kirillov et al., 2020; Yuan et al., 2020). Mask refinement methods often still operate at a reduced resolution or depend on features extracted with networks less powerful than standard segmentation networks. Sparse refinement methods are typically lower-quality than mask refinements (Tang et al., 2021; Kirillov et al., 2020; Yuan et al., 2020), or are explicitly designed for efficiency rather than high-quality segmentation (Liu et al., 2024). The gap in refinement quality can be explained since sparse methods trade coverage for efficiency, and they are therefore unable to correct catastrophic errors. However, recent advancements in image segmentation have significantly improved the quality of coarse masks, which brings into question the necessity of computationally expensive mask-level refinement. Indeed, we present a sparse refinement method for high-quality mask refinement that outperforms mask-based methods.

Sparse refinement methods select *where* to refine and determine *what* the refined values should be. Our core novelty is a policy called Boundary Adaptation (BA) for determining *where* to refine instance segmentation masks. BA iteratively re-classifies the mask boundary, as depicted in Figure 1. The motivation for BA comes from a set of empirical observations about its oracle case, in which each pixel selected for refinement is updated to its correct segmentation value. Each BA iteration is sparse: only about 0.1% to 1.5% of the image pixels are refined per step. BA is accurate: in some

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

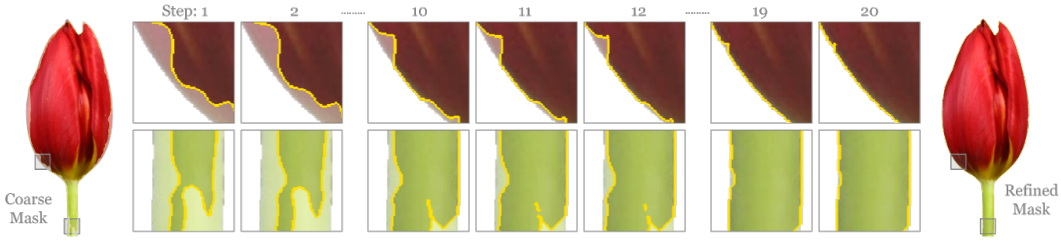


Figure 1: **Boundary Adaptation.** Boundary Adaptation is our novel pixel selection method for sparse segmentation refinement. The oracle method is applied to a coarse mask produced by SAM on an image from the ThinObjects-5K dataset (Liew et al., 2021). The pixels along the boundary of the segmentation mask (yellow) are iteratively re-classified using the ground-truth segmentation mask. Our specialized update rules presented in Section 4.1 ensure the mask topology remains coherent (no new disconnected regions) and the boundary remains smooth.

cases, only 5% of the total image pixels need to be queried to identify and correct nearly all of the erroneous mask values (see Section 3 for details).

To assess the value of BA beyond the oracle case, we also introduce the Boundary-SAM (B-SAM) architecture. Similar to related works (Kirillov et al., 2020), the B-SAM architecture produces refined mask values with a NeRF-like decoder using the current mask, pixel position, and low-level image features (Mildenhall et al., 2021). Compared to the state-of-the-art refinement methods (which all happen to be masked-based methods), B-SAM produces the highest quality masks on natural image datasets, a remote sensing dataset, and a CT scan dataset.

2 Related Works

Masked-based refinement methods can be broadly categorized by the resolution at which they operate: methods that predict masks at a fixed, low resolution, and methods that operate at or near the native image resolution.

Low-Resolution Mask Refinement. The dominant paradigm for segmentation refinement predicts a complete mask at a reduced resolution, which restricts refinement capability on high-resolution images, particularly for small objects. SAM (Kirillov et al., 2023) resizes inputs to 1024×1024 and produces masks at 256×256, necessarily blurring fine boundary detail. SAMRefiner (Lin et al., 2025) re-purposes the iterative mask refinement procedure used during SAM's own training, applying it at test time by engineering prompts derived from a coarse mask. This achieves strong results but inherits SAM's fixed-resolution constraint. HQ-SAM (Ke et al., 2023) attaches learnable modules to frozen SAM weights and trains on high-quality annotations, improving mask precision without resolving the fundamental resolution bottleneck. SegRefiner (Wang et al., 2023) takes a diffusion-inspired approach, iteratively denoising the mask itself, but likewise predicts at fixed resolution. A shared limitation is that upsampling from a low-resolution prediction degrades boundary quality and disproportionately harms small objects, where boundary pixels constitute a large fraction of the total mask area. In addition, the SAM-based methods (Lin et al., 2025; Ke et al., 2023) utilize the previous mask only through the prompt encoder, which requires bilinearly downsampling the mask to a 256×256 input, losing details from the prompt itself before refinement even begins.

Native-Resolution Mask Refinement. A few methods sidestep fixed-resolution prediction by operating over the image at full or near-native resolution. CascadePSP (Cheng et al., 2020) runs a segmentation network over local crops of the input image, which produces regular boundaries. However, this requires dozens of independent forward passes, making it slow in practice. CRM (Shen et al., 2022) avoids this cost by using a NeRF-like (Mildenhall et al., 2021) decoder that independently queries each pixel coordinate and can be batched efficiently, but because each point is decoded without local spatial context there is no mechanism to enforce boundary coherence across neighboring pixels. Compounding this, CRM's multiscale ResNet encoder is weaker than SAM-based features and aggressively downsamples spatial detail, disproportionately harming small objects.

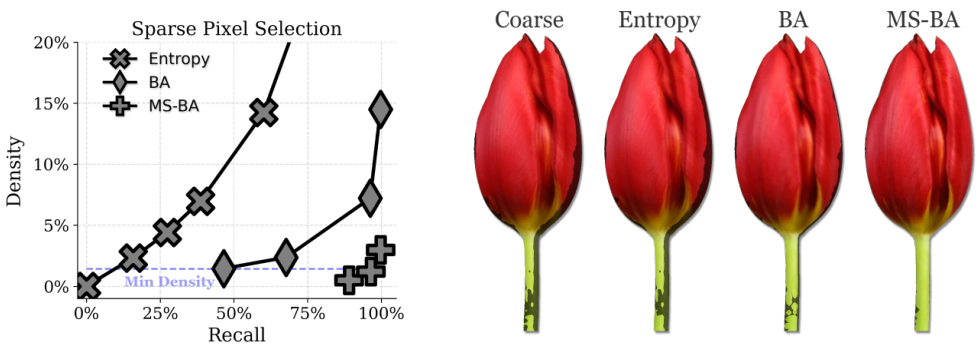


Figure 2: **Oracle Sparse Point Selection.** Sparse refinement methods trade-off the percent of total pixels that are queried (density) and the number of queried pixels that require correction (recall). (Left) For each refinement method, we evaluate a grid of hyperparameters on a coarse mask from SAM. (Right) A cropped region of the segmented object (a tulip) is visualized for comparison. The ground-truth segmentation is visualized as the semi-transparent background behind each tulip.

Sparse Refinement. Sparse methods refine fewer points than mask-based methods for computational efficiency, but and produce lower-quality refined masks. Related works for sparse refinement can be categorized by how the sparse set of pixels are selected for refinement. Examples include selecting pixels with high entropy (Huang et al., 2019; Kirillov et al., 2020; Liu et al., 2024), pixels along the mask boundary (Yuan et al., 2020), mask values that were estimated via interpolation (Ke et al., 2022), or a sparse selection of patches (Wu et al., 2020; Tang et al., 2021; Verelst & Tuytelaars, 2022). Here, we only consider entropy-based selection as a direct competitor to our proposed Boundary Adaptation. The method that refines pixels along the mask boundary can be seen as a single step of our method, and can produce speckling artifacts. The method that refines at interpolated pixel values is specialized to a specific architecture and assumes the initial low-resolution estimate is correct, an assumption that is not suitable for high-resolution refinement. Patch-based methods introduce an excessive computational burden, at which point tiling a high-quality segmentation network across the high-resolution image is a comparable alternative (Huang et al., 2018).

3 A Motivating Observation

Previous works have show that segmentation errors are concentrated along the boundary (Yuan et al., 2020; Ke et al., 2022), and our investigation begins with a related observation that iteratively re-classifying the mask boundary to the ground-truth label significantly improves mask quality. Figure 1 illustrates this oracle case applied to a coarse mask from SAM. Our core contribution is designing a refinement algorithm from this observation. We name our iterative boundary re-classification algorithm Boundary Adaptation (BA), and label its multi-scale version as MS-BA.

Understanding the significance of our observation requires context, so we compare the oracle BA/MS-BA methods to the well-known entropy-based method for sparse pixel selection, which we refer to as Entropy methods (Kirillov et al., 2020; Liu et al., 2024). Using logits from SAM, the Entropy method selects pixels for refinement if the pixel's entropy is larger than some threshold. The entropy values are computed by feeding the coarse mask into the SAM network.

A sparse selection strategy should query all pixels that require correction (high recall) while querying as few pixels as possible (low density/high precision). Sparse methods trade-off recall and density implicitly using a hyperparameter: an entropy threshold for Entropy methods or number of iterations for BA. Figure 2 (left) shows that compared to Entropy, the BA/MS-BA methods query significantly fewer pixels to achieve a significantly higher recall when both are endowed with oracle value updates to the selected pixels. In particular, MS-BA queries less than 5% of the total pixels to achieve a nearly perfect recall. Figure 2 (right) qualitatively compares the tulip masks, and we highlight that the stem of the tulip is significantly better for the BA/MS-BA methods compared to Entropy.

4 METHOD

4.1 BOUNDARY ADAPTATION

This section presents Boundary Adaptation (BA), our proposed method to select pixels for sparse refinement that iteratively re-classifies the pixels along the mask boundary. BA is inspired by the empirical observations in Section 3, but we include a theoretical understanding of BA in Section A.

We are given an image $\boldsymbol{x}$ of size $H \times W \times 3$ and a per-object binary mask $\boldsymbol{s}_0$ of size $H \times W$, where H and W are the native-resolution height and width of $\boldsymbol{x}$. Assuming the coarse mask $\boldsymbol{s}_0$ identifies each connected component, one can construct a sequence of intermediate masks that eventually match the ground-truth mask $\boldsymbol{s}^*$ by updating only pixels in the boundary $\mathcal{B}(\boldsymbol{s}_t)$ at each step; i.e., $\boldsymbol{s}_0 \to \boldsymbol{s}_1 \to \cdots \to \boldsymbol{s}_T = \boldsymbol{s}^*$, where T is the number of BA steps. A pixel i is considered part of the boundary if any of its 4-connected neighbors have a different label under the current mask $\boldsymbol{s}_t$.

Training Objective. In BA, a deep neural network DNN_θ predicts logits over boundary pixels,

$$\boldsymbol{z}_{t,i} = \text{DNN}_\theta(\boldsymbol{x}, \boldsymbol{s}_t, i), \qquad i \in \mathcal{B}(\boldsymbol{s}_t), \tag{1}$$

with classification probability $\widehat{p}_\theta(\boldsymbol{s}_{t+1,i} = 1 \mid \boldsymbol{s}_t, \boldsymbol{x}) = \sigma(\boldsymbol{z}_{t,i})$ where $\sigma(\cdot)$ is the sigmoid function. Since the ground-truth target is available at each time step of training, we can simply train a network using maximum likelihood estimation. Since only pixels in the boundary change between steps, which is about 0.1% to 1.5% of the $H \times W$ pixels, a network that operates sparsely can be trained at the native image resolution, even for ultra-high-resolution inputs exceeding 8000×8000 pixels. In addition to training on boundary points, we find the network quality improves by randomly sampling additional training points. The training objective combines a boundary loss with an auxiliary loss over a set $\mathcal{R}_t$ of points sampled uniformly across the image at each refinement step t:

$$L(\theta) = \sum_{t=0}^{T} \left[\sum_{i \in \mathcal{B}(\boldsymbol{s}_t)} \text{FocalBCE}(\sigma(\boldsymbol{z}_{t,i}),\, \boldsymbol{s}_i^*) + \lambda \sum_{i \in \mathcal{R}_t} \text{FocalBCE}(\sigma(\boldsymbol{z}_{t,i}),\, \boldsymbol{s}_i^\star) \right], \tag{2}$$

where FocalBCE is the focal binary cross-entropy loss function (Lin et al., 2017), and $\lambda = 0.5$ balances the two terms.

We produce our coarse segmentations synthetically, matching related works (Yuan et al., 2020; Cheng et al., 2020; Shen et al., 2022). These deformations corrupt the boundary of the ground-truth mask by randomly dropping/adding chunks of the contour, and randomly adding holes and disconnected mask points. Since the corruptions are random, the minimum value of T varies for each sample, but in our setup masks are mostly correct after about 6 iterations so we set $T = 6$ during training.

Inference Procedure. Naively replacing classified boundary pixels using the most likely mask value predicted by DNN_θ produces disconnected regions which appear as speckling artifacts. Figure 3 illustrates the possible speckling artifacts that naive BA can produce under a noisy oracle. Methods that operate pointwise are prone to this type of error (Yuan et al., 2020; Kirillov et al., 2020). In segmentation networks, extensive training and the low-resolution prediction itself will prevent these types of artifacts from forming in the first place.

Following the superpixel literature (Freifeld et al., 2015; Uziel et al., 2019), the solution to these speckling artifacts is to include a Potts term, which is a probabilistic prior that simultaneously restricts boundary updates to legal moves only and regularizes the boundary shape. This change splits the generative model of the segmentation labels into a two-step process. First, temporary labels are sampled from the prior $\tilde{\boldsymbol{s}} \sim p(\tilde{\boldsymbol{s}})$ Second, the actual labels are the result of applying a random permutation matrix, $\boldsymbol{s} = P\tilde{\boldsymbol{s}}$ with $p(\boldsymbol{s}|\tilde{\boldsymbol{s}}) \propto \mathbb{1}_{\text{valid}}[\boldsymbol{s}] \exp\left\{-\beta \sum_{j \sim i} \mathbb{1}_{\boldsymbol{s}_i \neq \boldsymbol{s}_j}(\boldsymbol{s})\right\}$, where the sum is over the 4-connected j neighbors of pixel i and the β term regularizes the complexity of the boundary. The indicator function ensures only topologically desirable updates are allowed. By re-interpreting the output of the network as the logits over temporary labels, including the Potts term only requires

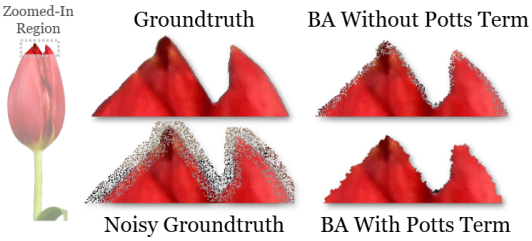


Figure 3: **The Potts Term Removes Speckling.** Naively assigning boundary pixels their refined values can produce speckling artifacts.

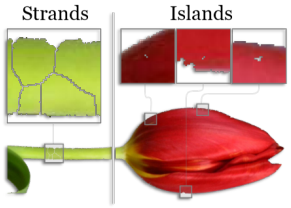


Figure 4: **Strands and Islands.** Topologically preserving updates produce strands and islands. Between BA steps, strands and islands are removed.

adding the log-Potts term logits to the network logits:

$$\log \frac{p_\theta(\boldsymbol{s}^i_{t+1}=1|\boldsymbol{s}_t,\boldsymbol{x})}{p_\theta(\boldsymbol{s}^i_{t+1}=0|\boldsymbol{s}_t,\boldsymbol{x})} = \log \frac{p_\theta(\tilde{\boldsymbol{s}}^i_{t+1}=1|\boldsymbol{s}_t,\boldsymbol{x})p(\boldsymbol{s}^i_{t+1}=1|\tilde{\boldsymbol{s}}^i_{t+1}=1,\tilde{\boldsymbol{s}}^{i\prime}_{t+1}=\boldsymbol{s}_t)}{p_\theta(\tilde{\boldsymbol{s}}^i_{t+1}=0|\boldsymbol{s}_t,\boldsymbol{x})p(\boldsymbol{s}^i_{t+1}=0|\tilde{\boldsymbol{s}}^i_{t+1}=0,\tilde{\boldsymbol{s}}^{i\prime}_{t+1}=\boldsymbol{s}_t)} \tag{3}$$

$$= \underbrace{\log \frac{p_\theta(\tilde{\boldsymbol{s}}^i_{t+1}=1|\boldsymbol{s}_t,\boldsymbol{x})}{p_\theta(\tilde{\boldsymbol{s}}^i_{t+1}=0|\boldsymbol{s}_t,\boldsymbol{x})}}_{\text{Logits, } z_{t,i}} + \underbrace{\log \frac{p(\boldsymbol{s}^i_{t+1}=1|\tilde{\boldsymbol{s}}^i_{t+1}=1,\tilde{\boldsymbol{s}}^{i\prime}_{t+1}=\boldsymbol{s}_t)}{p(\boldsymbol{s}^i_{t+1}=0|\tilde{\boldsymbol{s}}^i_{t+1}=0,\tilde{\boldsymbol{s}}^{i\prime}_{t+1}=\boldsymbol{s}_t)}}_{\text{log-Potts term}} \tag{4}$$

The Potts term can be efficiently computed using a Pytorch-friendly CUDA kernel in less than a millisecond. Notably, the next state mask updates for pixel i now depends on its neighbors. However, since the connectivity is restricted to only immediate neighbors, one can update one fourth of the pixels in parallel so all updates are complete after just four iterations.

While our primary interest in the Potts term is in preserving the original mask topology, the regularization coefficient β can also enforce smooth boundaries. While $\beta = 0$ in our experiments, in Section 5.3 we demonstrate that increasing β can significantly improve the refinement quality by encouraging the boundary to be smooth.

Heuristic Exceptions: Strands and Islands. While the Potts term defines our procedure for standard boundary updates, there are a few exceptions we handle heuristically. Specifically, single pixel strands or islands of background/foreground regions can occur because the Potts term preserves the coarse mask topology. Figure 4 illustrates these artifacts under the oracle case of BA. Periodically, we check for these single strands and remove them. This does permit the disconnection of two topologically connected regions, but in practice we find this happens rarely and only separates larger mask blobs, so the speckling artifacts we aimed to avoid still do not occur.

Multiscale Boundary Adaptation (MS-BA). BA requires chaining classification predictions across iterations, which can lead to a domain shift over time due to compounding model error. This leads to a drop in network accuracy, which in turn leads to irregular boundaries. In addition, it is desirable for BA to refine using fewer iterations for computational efficiency. For these two reasons, we designed a multiscale BA algorithm. To run the multiscale method, we first apply BA to the current mask at a lower resolution, constructed by bilinearly downsampling by some scaling factor (4, 2, or 1) and then binarizing via thresholding at 0.5. BA steps are executed at that particular scale, and when the scale change the refined mask is bilinearly upsampled to the next size. These initial coarse-scale refinement steps can achieve larger-scale corrections in fewer iterations while providing implicit regularization of the boundary. Final refinement steps at full resolution can then recover fine details without overfitting.

4.2 BOUNDARY-SAM

Architecture. BA determines where to refine, and the Boundary-SAM (B-SAM) pointwise decoder determines what the refined values should be. Algorithm 1 summarizes how B-SAM is used within the BA procedure. The B-SAM architecture we present is just one of many possible networks, and the macro-level design is common in literature (Kirillov et al., 2020; Shen et al., 2022; Liu et al., 2024). Its purpose in this paper is simply to understand the value of BA for sparse point selection.

The distinguishing feature of our B-SAM architecture is how the network interacts with the SAM features. SAM (Kirillov et al., 2023) produces mask logits using a dot-product between the decoded mask features and a context token. At a high-level, B-SAM estimates refined logits by modulating the decoded SAM mask features and context token using the current mask state, image features, and positional information. We perform this modulation using FiLM layers (Perez et al., 2018). The base network we use in B-SAM is an augmented variation of SAM called HQ-SAM (Ke et al., 2023), and image features are taken from an early layer in the HQ-SAM image encoder. See Section B for additional details.

Training. Following related works, we use synthetically deformed ground-truth masks as the coarse masks for training (Yuan et al., 2020). The refined values used during training to update the mask boundary between BA updates are randomly selected to be either the ground-truth labels or the network's previous predictions. Using the previous output of B-SAM as its input reduces the mismatch between training and testing distributions, and randomly sampling the ground-truth values ensures the gradients remain informative at later BA steps. We train B-SAM for 100,000 iterations using the AdamW optimizer with learning rate of 10^{-4} and a batch size of 4 (Loshchilov & Hutter, 2019). The memory consumption depends on the size of the coarse mask's boundary, and typically ranges between 7 GB to 11 GB. Training takes about 24 hours on one NVIDIA RTX 6000 GPU. During training, we use multiscale levels $\{4, 2, 1\}$ with $\{3, 3, 3\}$ refinement steps. During inference, we increase the number of steps to $\{12, 9, 3\}$ to compensate for model error, as the network empirically requires more steps than the oracle case to meaningfully improve the mask.

Algorithm 1 MS-BA with B-SAM

Require: Low-level features $\boldsymbol{f}_l$, semantic features $\boldsymbol{f}_s$, context token $\boldsymbol{c}$, coarse mask $\boldsymbol{s}_{0,1}$, native image resolution (H, W), number of steps at each scale $\{T_4, T_2, T_1\}$

$\boldsymbol{s} \leftarrow \boldsymbol{s}_0$
for Scale $k \in \{4, 2, 1\}$ **do**
  $\boldsymbol{f}_{l,k} \leftarrow \gamma_k \boldsymbol{f}_l + \beta_k$ (Multiscale FiLM)
  $\boldsymbol{s}_k \leftarrow \texttt{resize}(\boldsymbol{s}, H/k, W/k)$
  for $t = 1$ **to** T_k **do**
    $\boldsymbol{z}^{\mathcal{B}(\boldsymbol{s}_{t,k})} \leftarrow \{\texttt{B-SAM}(\boldsymbol{f}_k,\ \boldsymbol{f}_s,\ \boldsymbol{c},\ \boldsymbol{s}_{t,k}, i)\}_{i \in \mathcal{B}(\boldsymbol{s}_{t,k})}$
    $\boldsymbol{s}_{t+1,k} \leftarrow \texttt{UpdateMask}(\boldsymbol{z}^{\mathcal{B}(\boldsymbol{s}_k)},\ \boldsymbol{s}_{t,k})$
  end for
  $\boldsymbol{s} \leftarrow \texttt{resize}(\boldsymbol{s}_k, H, W)$
end for
return $\boldsymbol{s}$

5 Experiments

We report the quality of the masks refined with B-SAM with BA is superior existing refinement methods on several natural image datasets (Section 5.1), a remote sensing dataset (Section 5.2), and a medical imaging dataset (Section 5.2). Section 5.3 includes a subset of our ablation studies that show the mask quality of BA is superior to using entropy for sparse pixel selection, validates the iterative nature of BA, validates the utility of multi-scale inference, and demonstrates the value of regularizing the mask boundary with the Potts term. Section C includes additional experiments such as a computational comparison that shows B-SAM is comparable to existing methods, refinement results when the coarse masks are produced with lower-quality networks, and an assessment of topological error vs model error.

Baselines. To assess sparse pixel selection for mask refinement, we compare BA/MS-BA with the Entropy selection method (Kirillov et al., 2020; Liu et al., 2024). To assess overall mask refinement quality, we compare with SAM-Refiner (Lin et al., 2025), the high-quality version of SAM-Refiner we call HQSAM-Refiner (Ke et al., 2023), CRM (Shen et al., 2022), and CPSP (Cheng et al., 2020).

Natural Image Datasets. We consider a mixture of high-resolution datasets with high-quality annotations, including ThinObject-5K (Liew et al., 2021), DIS5K (Qin et al., 2022), UHRSD (Xie et al., 2022), HRSOD (Zeng et al., 2019), and BIG (Cheng et al., 2020). These datasets are commonly used for benchmarking refinement methods. In these experiments, the MS-BA steps are set to $\{4 : 12, 2 : 9, 1 : 3\}$ where the number before the colon indicates the scale and the number after the colon indicates the number of refinement steps at that scale.

Table 1: **Refining SAM on Natural Image Datasets.** Comparison of segmentation quality across all natural image datasets, refining SAM masks created by prompting SAM with the ground-truth bounding box. Reported metrics are the mIoU / bIoU. Blue and orange coloring indicates the best and second best method according to the mIoU value, excluding the oracle case. The Oracle case executes five pixel-width boundary updates at the native resolution.

Dataset	Input (SAM)	Oracle BA	B-SAM (Ours)	HQSAM Refiner	SAM Refiner	CRM	CPSP
ThinObjects	85.13 / 73.52	91.49 / 85.39	**88.91 / 80.24**	87.74 / 79.96	**88.28 / 77.45**	81.06 / 68.34	87.58 / 79.24
HRSOD	90.14 / 80.97	93.81 / 89.90	**91.28 / 83.12**	**91.01 / 82.72**	90.06 / 81.49	89.69 / 80.33	90.84 / 83.26
UHRSD	87.17 / 74.76	90.75 / 83.83	**89.03 / 77.94**	**88.86 / 78.99**	86.98 / 76.07	88.17 / 76.87	88.52 / 79.49
BIG	84.46 / 75.09	88.70 / 83.86	**85.39 / 76.69**	85.32 / 75.23	84.92 / 76.10	84.72 / 74.41	**85.90 / 78.59**
DIS5K	66.86 / 58.16	74.63 / 68.15	**71.19 / 63.38**	**72.03 / 66.38**	67.38 / 58.79	69.47 / 62.08	69.96 / 62.80

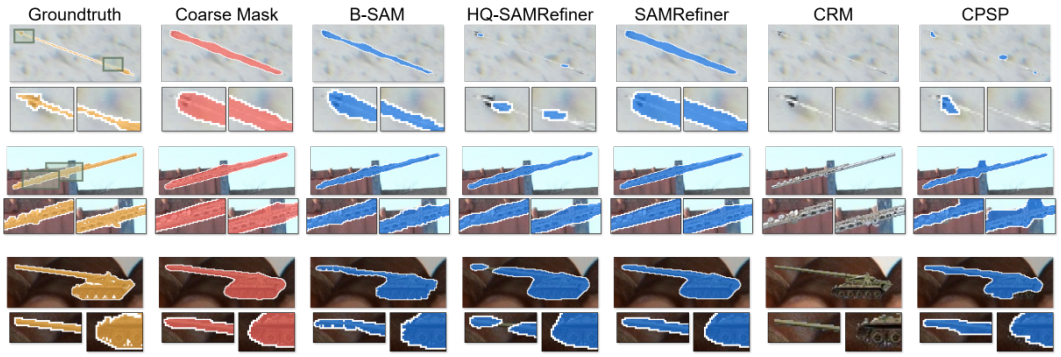


Figure 5: **Refining Segmentation Masks of Natural Images.** For several refinement methods (HQ-SAMRefiner, CRM, CPSP), the masks of small objects in high-resolution images are completely dropped. In contrast, BA is capable of refining masks of small objects.

Remote Sensing Dataset. Remote sensing datasets for segmentation tasks typically require recognizing small objects in high-resolution images. We consider the SegFVG dataset, which consists of segmentations of homes in Italy (Rota et al., 2025). A single SegFVG segmentation is associated with several ground-truth objects (houses). Since our method is designed to refine one object mask at a time, we split each connected component into a separate mask. We use MS-BA steps $\{1:9\}$.

Medical Imaging Dataset. This experiment compares segmentation refinement methods on kidney tumors from CT scans in the KITS 2021 dataset (Heller et al., 2021). The dataset consists of dense 3D voxel grids with CT-machine-specific spacing. To create a 2D image dataset, we extract slices along the z-axis that include kidney tumor segmentations larger than 10 total pixels, as smaller segmentations are less reliable. Disconnected ground-truth segmentations are treated as separate tumors. We use MS-BA steps $\{1:9\}$.

Evaluation Metrics. We assess segmentation mask quality by computing the mask and boundary intersection-over-union (mIoU / bIoU) between the candidate and ground-truth masks. The mIoU metric is computed by dividing the number of pixels in the intersection of the two binary masks by the number of pixels in their union. The bIoU metric is computed identically, but on boundary masks derived by dilating the candidate and ground-truth object boundaries by a width proportional to the square root of the object's area (Cheng et al., 2021).

5.1 Refinement on Natural Image Datasets

We compare B-SAM with recent works for high-resolution refinement on natural image datasets. The coarse masks are produced by prompting SAM Kirillov et al. (2023) with the ground-truth bounding boxes. Additionally, we construct a synthetic dataset to isolate the effect of object size, demonstrating that B-SAM is the only refinement method that consistently improves mask quality across small, medium, and large objects. Section C.1 includes refinement results when the coarse masks are produced using an older, lower-quality network, a comparison computational costs, and an assessment of topological vs model error.

Table 2: **Refining SAM on Remote Sensing and Medical Imaging Datasets.** Comparison of segmentation quality on the SegFVG and KITS2021 datasets when refining SAM masks created by prompting SAM with the ground-truth bounding box. Reported metrics are the mIoU / bIoU. Blue and orange coloring indicates the best and second best method according to the mIoU value, excluding the oracle case. The Oracle case executes five pixel-width boundary updates at the native resolution.

Dataset	Input (SAM)	Oracle BA	B-SAM (Ours)	HQSAM Refiner	SAM Refiner	CRM	CPSP
SegFVG	86.08 / 85.42	94.28 / 93.82	**87.41 / 86.81**	83.27 / 82.35	85.40 / 84.78	**87.27 / 86.65**	69.63 / 69.02
KITS 2021	81.10 / 74.78	98.15 / 96.68	**81.74 / 75.77**	73.37 / 62.88	75.44 / 67.50	**76.85 / 70.55**	71.58 / 66.09

Refining SAM. Table 1 shows that B-SAM is the best refinement method for masks produced by SAM. SAM(-huge) is prompted with the ground-truth bounding box for each object instance to produce the initial masks, which are then refined by each indicated method. As both SAMRefiner and HQ-SAMRefiner require interpolating a low-resolution mask, their ability to segment small objects in large images is fundamentally limited. In contrast, both CRM and Cascade-PSP operate at the native image resolution, but their lower-capacity networks often lead to lower-quality segmentations.

B-SAM Excels at Refining Small Objects. B-SAM's combination of SAM features and point-wise decoding make it particularly capable of segmenting small objects in large images. Since most objects in high-quality annotation datasets are relatively large, about 512×512 pixels in size, assessing the impact of object size with the original natural image datasets is not possible. Accordingly, we construct our own dataset to stratify by object size. Similar to how many of these datasets are originally created, we cut objects from one dataset and paste them into another image. In this case, we cut 400 random objects from ThinObjects-5K and paste them into 400 random images from UHSRD2K. The ThinObjects-5K dataset has the most detailed objects and UHSRD2K images have a consistent resolution of about 2k $\times$ 2k. The coarse masks are again produced from SAM prompted with the ground-truth bounding box. B-SAM's quality is the best overall among refinement methods, and B-SAM is the only refinement method to improve segmentation quality across all object sizes. Figure 5 shows example refined masks and Figure 6 shows the quality of each method stratified by object size.

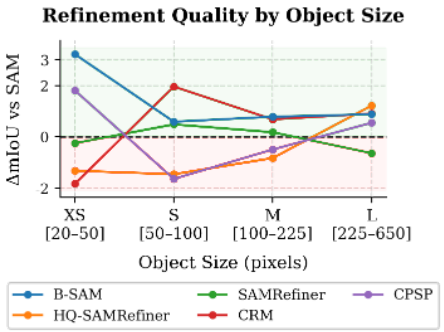


Figure 6: **B-SAM Refines Small Objects.** B-SAM is the best refinement network for small objects, and is the only method that consistent improves the mask quality.

5.2 Refinement on Remote Sensing and Medical Imaging Datasets

Natural image datasets are commonly used for benchmarking, but segmentation and segmentation refinement methods are commonly used in practice across other domains. Table 2 shows results in two such applications: remote sensing using the SegFVG dataset (Rota et al., 2025) and medical imaging using the KITS 2021 dataset (Heller et al., 2021). For a qualitative comparison, see Figure 9 in Section C.1. In these experiments, we re-train B-SAM, HQ-SAM (auxilary features), CRM, and CPSP on each new dataset, and modify the multiscale method of both B-SAM and CRM to only operate at the native resolution to better capture small objects. For B-SAM, we train with 5 refinement steps and test with 8 refinement steps. Training on the SegFVG dataset takes less than 12 hours for each method on a single NVIDIA A6000 GPU. raining on the KITS2021 dataset takes less than 10 hours for each method on a single NVIDIA A6000 GPU. In Table 2, B-SAM is the only method to improve segmentation quality on average.

5.3 Select Ablation Studies

Due to space constraints, only some of the ablation studies are presented in the main paper. Please refer to Section C.2 for additional experiments, which include a comparison with alternative sparse

decoding strategies (beyond Entropy), validating the utility of native-resolution decoding, and B-SAM architecture ablations.

BA vs Entropy. Our work was motivated by the oracle-case observation that BA/MS-BA pixel selection outperforms Entropy selection. Here, we validate this observation in practice by training a B-SAM network using entropy-based point selection using coarse masks and logits from the SAM network. Note that, since we evaluate on coarse masks derived from SAM, the train-test domain gap for this method is smaller than our standard training method for B-SAM, which uses synthetic coarse masks. Despite this advantage, on the ThinObjects dataset the Entropy-based B-SAM achieves mIoU/bIoU of 84.75 / 75.29, while BA-based B-SAM achieves 88.91 / 80.24.

Validating Iterative Boundary Updates. BA proposes refining a segmentation mask by iteratively updating the boundary, which is distinct from prior work that typically operates on either the entire image or patches surrounding the mask boundary. To demonstrate the value of iterative updates, we compare our full approach against a single-step baseline in which B-SAM is queried once across all image pixels to produce a mask directly. Using just this single-step output yields a mIoU/bIoU of 87.32 / 77.66 on ThinObjects-5K, a significant degradation compared to our iterative approach which achieves 88.91 / 80.24. This gap shows that the quality of B-SAM's predictions depend on both the network itself and on the iterative correction procedure that allows errors to be progressively resolved over multiple steps.

Validating Multiscale Inference. Taking more BA steps can increase the chance of error, since model errors can compound over time and cause a distribution shift in the input mask. The multiscale approach mitigates this by covering a larger boundary area per step, reducing the total number of steps required, and limiting the opportunity for error accumulation. To verify this, we train a B-SAM network without the multiscale step and run 100 BA steps during inference at full resolution, where the step count was chosen to maximize mIoU. On ThinObjects-5K, this single-scale variant achieves a mIoU/bIoU of 86.32 / 74.98, significantly worse than the multiscale variant at 88.91 / 80.24. The especially large gap in bIoU suggests that multiscale inference is particularly important for producing clean, well-localized boundaries rather than just improving overall mask coverage. This is validated in Figure 10 in Section C.2.

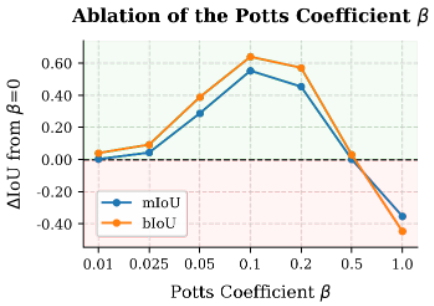


Figure 7: **Boundary Regularization.** Plotted is the change in IoU relative to $\beta = 0$ when varying the Potts coefficient on the BIG dataset.

Regularization via the Potts Term. While our iterative boundary update experiments implicitly validate the Potts term's topology-preserving indicator function, the effect of the regularization coefficient β warrants separate investigation. Figure 7 shows mask quality on the BIG dataset across a grid of β values. Both mIoU and bIoU peak at $\beta = 0.1$ (85.94 / 77.33), placing B-SAM first on the BIG dataset (Table 1), suggesting that boundary regularization is a valuable component of BA. As β grows further, both metrics decline as the regularization penalty increasingly inhibits meaningful mask updates. For simplicity, we set $\beta = 0$ in all other experiments, retaining only the topology-preserving indicator function to prevent speckling artifacts.

6 Discussion

We proposed Boundary Adaptation (BA), a policy for selecting sparse pixels for mask refinement that iteratively re-classifies the mask boundary. Naively implementing BA can lead to speckling artifacts, improperly disconnected regions, and irregular boundaries; we address these problems by updating a network's output logits with a Potts term for regularization, and heuristically removing single-pixel artifacts. Our multi-scale variation of BA (MS-BA) also yields significant improvement over naive BA. To implement BA/MS-BA, we introduce the Boundary-SAM (B-SAM) architecture that operates at the native image resolution using frozen SAM features. On standard natural image datasets, our sparse refinement method (B-SAM + MS-BA) yields higher-quality masks than mask-based refinement methods. Our approach generalizes to specialized segmentation tasks in medical imaging and remote sensing, and yields substantial improvements especially for small objects.

7 MANDATORY

AI USE STATEMENT

In this work, we used generative AI tools for polishing an earlier draft of the paper and writing some of the code used to visualize the results. We have not used generative AI tools for any of the following tasks: generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Our primary benchmark results are the refinement results on natural image datasets in Table 1. Code to reproduce this table and execute B-SAM on 2D images will be made publicly available upon acceptance.

REFERENCES

Michela Antonelli, Annika Reinke, Spyridon Bakas, Keyvan Farahani, Annette Kopp-Schneider, Bennett A Landman, Geert Litjens, Bjoern Menze, Olaf Ronneberger, Ronald M Summers, et al. The medical segmentation decathlon. *Nature Communications*, 13(1):4128, 2022.

Patrick Bilic, Patrick Christ, Hongwei Bran Li, Eugene Vorontsov, Avi Ben-Cohen, Georgios Kaissis, Adi Szeskin, Colin Jacobs, Gabriel Efrain Humpire Mamani, Gabriel Chartrand, et al. The liver tumor segmentation benchmark. *Medical image analysis*, 84:102680, 2023.

Bowen Cheng, Ross Girshick, Piotr Dollár, Alexander C Berg, and Alexander Kirillov. Boundary iou: Improving object-centric image segmentation evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'21)*, pp. 15334–15342, 2021.

Ho Kei Cheng, Jihoon Chung, Yu-Wing Tai, and Chi-Keung Tang. Cascadepsp: Toward class-agnostic and very high-resolution segmentation via global and local refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'20)*, pp. 8890–8899, 2020.

Oren Freifeld, Yixin Li, and John W Fisher. A fast method for inferring high-quality simply-connected superpixels. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'15)*, pp. 2184–2188, 2015.

Vivien Sainte Fare Garnot and Loic Landrieu. Panoptic segmentation of satellite image time series with convolutional temporal attention networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR'21)*, pp. 4872–4881, 2021.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'17)*, pp. 2961–2969, 2017.

Nicholas Heller, Fabian Isensee, Klaus H Maier-Hein, Xiaoshuai Hou, Chunmei Xie, Fengyi Li, Yang Nan, Guangrui Mu, Zhiyong Lin, Miofei Han, et al. The state of the art in kidney and kidney tumor segmentation in contrast-enhanced ct imaging: Results of the kits19 challenge. *Medical Image Analysis (MedIA'21)*, 67:101821, 2021.

Bohao Huang, Daniel Reichman, Leslie M Collins, Kyle Bradbury, and Jordan M Malof. Tiling and stitching segmentation output for remote sensing: Basic challenges and recommendations. *arXiv preprint arXiv:1805.12219*, 2018.

Yu-Hui Huang, Marc Proesmans, Stamatios Georgoulis, and Luc Van Gool. Uncertainty based model selection for fast semantic segmentation. In *2019 16th International Conference on Machine Vision Applications (MVA'19)*, pp. 1–6. IEEE, 2019.

Lei Ke, Martin Danelljan, Xia Li, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Mask transfiner for high-quality instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*, pp. 4412–4421, 2022.

Lei Ke, Mingqiao Ye, Martin Danelljan, Yu-Wing Tai, Chi-Keung Tang, Fisher Yu, et al. Segment anything in high quality. *Advances in Neural Information Processing Systems (NeurIPS'23)*, 36: 29914–29934, 2023.

Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'20)*, pp. 9799–9808, 2020.

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 (ICCV'23)*, pp. 4015–4026, 2023.

Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'23)*, pp. 3041–3050, 2023.

Jun Hao Liew, Scott Cohen, Brian Price, Long Mai, and Jiashi Feng. Deep interactive thin object selection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV'21)*, pp. 305–314, 2021.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'17)*, pp. 2980–2988, 2017.

Yuqi Lin, Hengjia Li, Wenqi Shao, Zheng Yang, Jun Zhao, Xiaofei He, Ping Luo, and Kaipeng Zhang. Samrefiner: Taming segment anything model for universal mask refinement. In *The Thirteenth International Conference on Learning Representations (ICLR'25)*, 2025.

Zhijian Liu, Zhuoyang Zhang, Samir Khaki, Shang Yang, Haotian Tang, Chenfeng Xu, Kurt Keutzer, and Song Han. Sparse refinement for efficient high-resolution semantic segmentation. In *European Conference on Computer Vision (ECCV'24)*, pp. 108–127. Springer, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *The Seventh International Conference on Learning Representations (ICLR'19)*, 2019.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'18)*, volume 32, 2018.

Xuebin Qin, Hang Dai, Xiaobin Hu, Deng-Ping Fan, Ling Shao, and Luc Van Gool. Highly accurate dichotomous image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV'22)*, pp. 38–56. Springer, 2022.

Maryam Rahnemoonfar, Tashnim Chowdhury, Argho Sarkar, Debvrat Varshney, Masoud Yari, and Robin Roberson Murphy. Floodnet: A high resolution aerial imagery dataset for post flood scene understanding. *IEEE Access*, 9:89644–89654, 2021.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241. Springer, 2015.

Claudio Rota, Flavio Piccoli, Rajesh Kumar, and Gianluigi Ciocca. A high-resolution large-scale dataset for building segmentation from aerial imagery in northeastern italy. *Scientific Data*, 12 (1):1735, 2025.

Tiancheng Shen, Yuechen Zhang, Lu Qi, Jason Kuen, Xingyu Xie, Jianlong Wu, Zhe Lin, and Jiaya Jia. High quality segmentation for ultra high-resolution images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*, pp. 1310–1319, 2022.

Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'19)*, 2019.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'20)*, June 2020.

Chufeng Tang, Hang Chen, Xiao Li, Jianmin Li, Zhaoxiang Zhang, and Xiaolin Hu. Look closer to segment better: Boundary patch refinement for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'21)*, pp. 13926–13935, 2021.

Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *European Conference on Computer Vision (ECCV'20)*, pp. 282–298. Springer, 2020.

Roy Uziel, Meitar Ronen, and Oren Freifeld. Bayesian adaptive superpixel segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV'19)*, pp. 8470–8479, 2019.

Thomas Verelst and Tinne Tuytelaars. Segblocks: Block-based dynamic resolution networks for real-time segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI'22)*, 45(2):2400–2411, 2022.

Ao Wang, Hui Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Lsnet: See large, focus small. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR'25)*, pp. 9718–9729, 2025.

Mengyu Wang, Henghui Ding, Jun Hao Liew, Jiajun Liu, Yao Zhao, and Yunchao Wei. Segrefiner: Towards model-agnostic segmentation refinement with discrete diffusion process. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS'23)*, pp. 79761–79780, 2023.

Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. In *European Conference on Computer Vision (ECCV'20)*, pp. 649–665. Springer, 2020.

Tong Wu, Zhenzhen Lei, Bingqian Lin, Cuihua Li, Yanyun Qu, and Yuan Xie. Patch proposal network for fast semantic segmentation of high-resolution images. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'20)*, volume 34, pp. 12402–12409, 2020.

Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*, pp. 3974–3983, 2018.

Chenxi Xie, Changqun Xia, Mingcan Ma, Zhirui Zhao, Xiaowu Chen, and Jia Li. Pyramid grafting network for one-stage high resolution saliency detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*, pp. 11717–11726, 2022.

Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*, pp. 2970–2979, 2017.

Yuhui Yuan, Jingyi Xie, Xilin Chen, and Jingdong Wang. Segfix: Model-agnostic boundary refinement for segmentation. In *European Conference on Computer Vision (ECCV'20)*, pp. 489–506. Springer, 2020.

Yi Zeng, Pingping Zhang, Jianming Zhang, Zhe Lin, and Huchuan Lu. Towards high-resolution salient object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV'19)*, 2019.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems*, 2023.