A. Code Availability

All necessary code to reproduce PromptMoE, our DES model weights, and all experimental results can be found here: https://github.com/sprice134/PromptMoE.

B. Acronyms and Abbreviations

In this work, we use a variety of acronyms and abbreviations, as defined in Tab. 4.

Table 4. Summary of acronyms and notation used throughout the paper.

Symbol / Acronym	Description
IIP	Image-Informed Prompting (Sec. 3.1)
DES	Dynamic Expert Selector (Sec. 3.2)
PPE	Prompt-Placement Explorer (Sec. 3.3)
x	Image
M_0	Initial (unrefined) coarse mask
$H \times W$	Image resolution / size of x and M_0
$\hat{M}$	Output (refined) mask
$\mathcal{E}$	Set of all experts
v_{e_i}	Learned embedding for expert e_i
$\mathbf{V}$	Expert embedding matrix (all v_{e_i})
h	Joint context vector from (x, M_0)
d_{ctx}	Dimension of the context vector h
ϕ_{e_i}	Feature vector for utility prediction
U_{e_i}	Predicted utility of expert e_i
ψ_{e_i,e_j}	Pairwise feature vector for interaction
I_{e_i,e_j}	Predicted interaction of experts e_i and e_j
S_{e_i,e_j}	Combined score of expert pair (e_i, e_j)
$(e_{i^\star}, e_{j^\star})$	Selected optimal pair of experts
$w_{i^\star}, w_{j^\star}$	Mixture weights for fusing expert maps
r_{e_i}	Relevance map from expert e_i, $[0,1]^{H\times W}$
r_{e_i,e_j}	Fused relevance map from e_i and e_j, $[0,1]^{H\times W}$
Ω	Pixel coordinate space ($H \times W$)
E_t	Excluded pixels for prompt placement at step t
M_t	Eligible pixels (subset of M_0) at step t
λ	Suppression factor for PPE
ρ_t	Adaptive suppression radius at step t
IoU	Intersection-over-Union
BIoU	Boundary IoU (2% of image diagonal)
ΔIoU/BIoU	Change in IoU/BIoU vs. unrefined mask
$\Delta\Delta$IoU/BIoU	Change in ΔIoU/BIoU vs. another method

C. Vision Tasks

We target three segmentation settings with this work (semantic, instance, and salient object segmentation), as shown in Figure 8. While each task focuses on determining the perimeter of objects, each approach has its own unique challenges. Semantic segmentation seeks to assign a class label to every pixel (e.g., background, person, dog), but does not attempt to distinguish multiple instances of the same class. In contrast, instance segmentation delineates and labels each object instance separately. Lastly, salient object segmentation outputs a class-agnostic binary foreground mask that highlights the most visually prominent object(s). The foreground may span multiple classes and is not required to be exhaustive.

In Fig. 8, for semantic segmentation, the four individuals would be grouped together as the *person* class, while the dog is labeled *dog*, and the remainder is labeled *background*. In instance segmentation, the dog would have the same label (*dog*), since there was only one instance, while the person class is separated into four distance *person* objects. Lastly, for salient segmentation, the *person* and *dog* objects are all merged to form a *foreground* label. We seek a single, robust segmentation refinement system that can handle each of these tasks, and their unique failure modes.

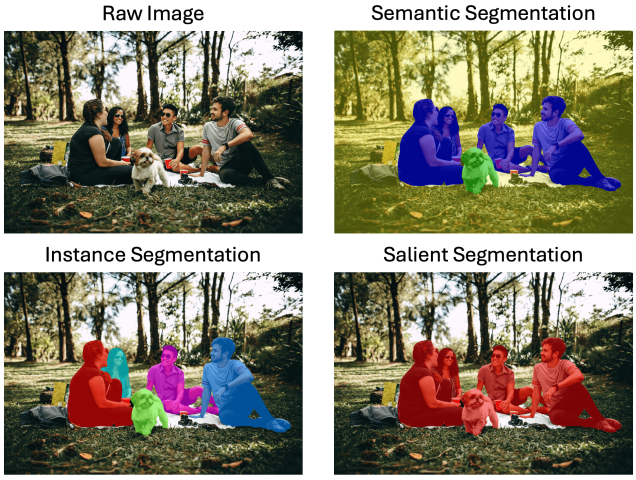

Figure 8. Illustration of the three target domains. Top-left: input image; top-right: semantic segmentation (per-pixel class labels); bottom-left: instance segmentation (separate object instances); bottom-right: salient object segmentation (binary foreground mask).

D. Societal and Ethical Impacts

PromptMoE is a general-purpose segmentation refinement framework. As a result, the societal and ethical impacts are dependent on its specific application. However, we have identified three primary areas of potential negative impact:

1. **Surveillance:** By improving the precision of segmentation, PromptMoE could be used to enhance surveillance systems, enabling more accurate and persistent tracking of individuals or objects.
2. **Photo/Video Manipulation:** While PromptMoE can accelerate and improve photo-editing tasks like background removal or object selection, the same capabilities could be used to facilitate photo and video manipulation (e.g. “deepfakes”) with precise object extraction.
3. **Computational Costs:** PromptMoE utilizes SAM, a large-scale foundational model that required significant

computational power to train and introduces additional computational costs over an initial prediction without refinement. This could result in additional energy/resource consumption.

We attempt to mitigate these computational impacts with our DES router, preventing dense evaluations for each image, PromptMoE-Lite, restricting the most computationally expensive experts, and demonstrating successful refinement capabilities with smaller, less expensive backbones. Additionally, we believe that while the potential negative impacts with respect to surveillance and photo manipulation are significant, they are not unique to our specific method. These same risks would be true of any general purpose vision model.

E. Key Challenges Addressed by PromptMoE

In this work, we address four challenges faced by promptable segmentation refinement frameworks (Sec. 1). Here, we provide more in-depth visualizations for each, as depicted below in Fig. 9, 10, 11, and 12.

C1. Semantic Ambiguity: For promptable vision models like SAM, they seek to maximize prompt alignment rather than a specific class. As a result, when placing a prompt at the center of an object, slight variations can have drastic effects. For example, in Fig. 9, a minor perturbation in prompt location (less than 5% of the image diagonal), can result in detecting the jacket, shirt, tie, or whole person. As a result, complex, well-designed prompts are often required to ensure accurate downstream segmentations.

C2. Intra-Prompt Interaction: Even when each prompt component is individually well placed, their joint effect must also be considered. As shown in Fig. 10, three positive points (Prompt A, B and C) each yield nearly identical, high-quality segmentations of the car. However, when Prompt A + B + C are used, the final segmentation collapses to the specific subregion containing this points, suppressing other true subregions of the object, degrading refinement performance. A similar effect can be seen in the flower, where single-point prompts yield high-performing performance, but when used in combination, degrade the mask to a single petal. As a result, when building an automated prompt generator, users must use a sufficiently complex prompt to inform the segmentor about the desired object (addressing C1), but also consider the intra-prompt effect to prevent expert collapse (addressing C2). In this work, we present our Prompt-Placement Explorer (PPE), enabling us to increase the number of points while ensuring high-confidence *and* spatially diverse prompts to address both challenges.

C3. Noisy Inputs: Coarse masks produced by a base predictor are noisy, often containing structural errors such as holes, missing parts, or over-filled regions. As a result, an effective refiner must be able to identify these noisy regions and suppress them when constructing a prompt for refine-

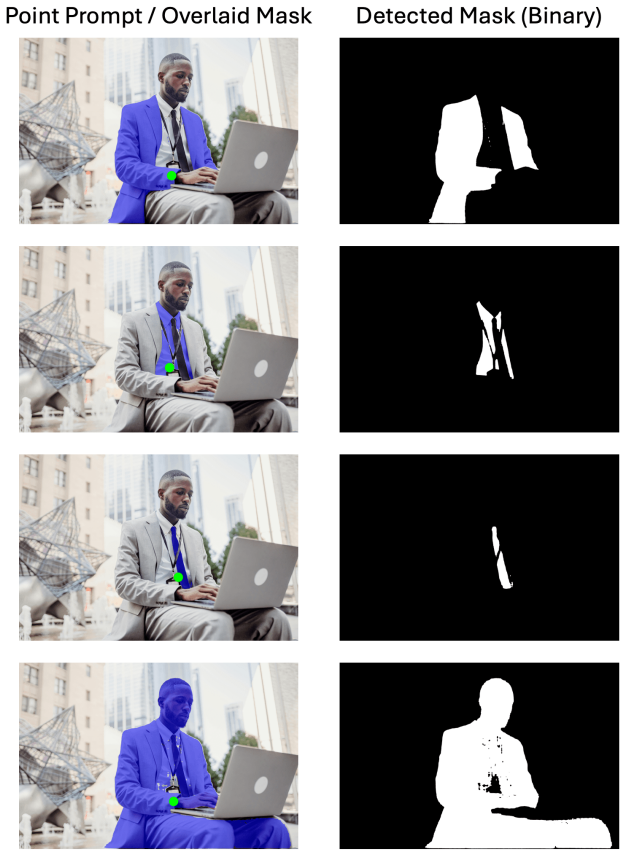

Figure 9. **Challenge C1: Semantic Ambiguity**. Four single-point prompts placed near each other producing drastically different segmentations, identifying the jacket, shirt, tie, or entire person.

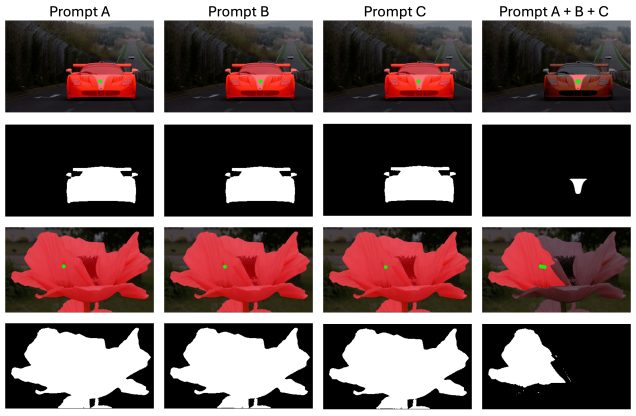

Figure 10. **Challenge C2: Intra-prompt Interaction**. Three nearby positive prompts (A-C) each produce almost identical, correct masks when used individually (left three columns), but combining them (right) overemphasizes their shared region and suppresses the remainder of the object, leading to a severely degraded segmentation.

ment. For example, in Fig. 11, the initial mask of a donut has the center incorrectly filled. An edge-distance heuristic, which assumes interior pixels are the safest, assigns highest confidence to this erroneous center, incorrectly placing a prompt in the middle of the hole, degrading the final mask quality. In contrast, our PromptMoE framework using a mixture-of-experts, correctly down-weighted this hole, placing multiple prompts in the true donut, resulting in an improved final mask.

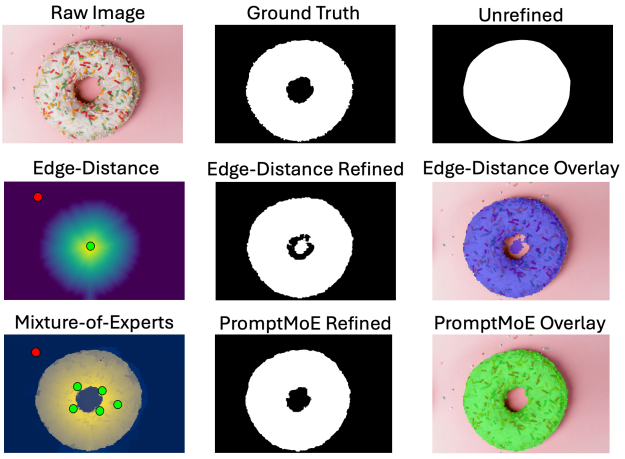

Figure 11. **Challenge C3: Noisy Inputs**. A coarse mask of a donut with a filled-in hole illustrates how noise in the input can leak into the prompt if not handled correctly. An edge-distance heuristic treats the erroneous interior as high confidence and places a false inclusion point, degrading the final mask (row 2). In contrast, PromptMoE fuses multiple image-processing cues to suppress this noise and place prompts along the true donut boundary (row 3).

C4. Diverse Image Characteristics: Previous works, such as SAMRefiner [29] and DualSight [35], have used fixed heuristics to place point prompts. Namely, SAMRefiner places a single point at the center of an object, where it is least likely to be incorrect, and DualSight seeks to maximize distance between multiple points, pushing points towards the boundary. In this work, we demonstrate that a single fixed heuristic is insufficient for robust refinement. For example, in Fig. 12, we highlight how varying experts can provide highly informative or even misleading guidance for prompting.

In the first row of Fig. 12, a warning sign on the beach, the edge-distance map correctly gives low confidence to the surrounding sand, high confidence to pixels on the sign, and the highest confidence near the center of the sign, where a point prompt would be most reliable. Depth-similarity provides no distinction between the sign and beach, as expected since there is no real depth change between the sign and background. In contrast, color-similarity actually provides a *negative* signal, assigning higher confidence to the sand than to the sign. This likely occurs because the sign is white with black letters, so the mean color of the mask (M_0) is closer to the sand than to either the white or black on the sign.

In the second row of Fig. 12, a butterfly sitting on a flower, edge-distance and depth-similarity both perform well, giving high confidence to the foreground and low confidence to the surrounding plants. Texture-similarity also gives a useful signal, assigning its highest scores to parts of the flower and butterfly. However, some regions of the true object receive very low texture scores, indicating that this cue exhibits high specificity (many background pixels are suppressed) but relatively low sensitivity (some foreground pixels are missed).

In the third row of Fig. 12, a chair scene, depth- and color-similarity provide strong cues, cleanly separating the blue chairs from the green background. Texture correctly down-ranks a large portion of the grass relative to the chairs, but fails to suppress all background regions. While edge-distance is capable of separating the chair from the background that was not included in the intial detection, it fails to detect the spurious over-segmentations in the back of the chair such that many of the highest confidence pixels from edge-distance are actually incorrect.

F. Image-Informed Prompting (IIP) Experts

In this work, we present Image-Informed Prompting (IIP), a mixture-of-experts framework with ten image-processing experts. We provide a description of each, as well as any relevant equations below:

F.1. Noise-Tolerant Prompting

- **Edge Distance.** Computes the Euclidean distance of each pixel p in M_0 to the closest boundary of M_0, then normalizes by the maximum distance so interior pixels receive higher scores and pixels near the boundary receive lower scores:

$$d_{M_0}(p) := \mathrm{EDT}(M_0, p) \tag{10}$$

$$\mathcal{E}_{\text{edge}}(M_0, p) = \frac{dM_0(p)}{\max\limits_{u \in M_0} d_{M_0}(u)} \tag{11}$$

F.2. Context-Aware Prompting Experts

- **Color Similarity.** Computes the mean color ($\bar{C}$) over M_0 and scores pixels by how close they are to it, where pixels whose color is closer to $\bar{C}$ are scored closer to 1.

$$\bar{C} = \frac{1}{|M_0|} \sum_{u \in M_0} C(x, u) \tag{12}$$

$$\mathcal{E}_{\text{color}}(x, p) = 1 - \frac{\left|C(x,p) - \bar{C}\right|2}{\max u \in M_0 \left|C(x,u) - \bar{C}\right|_2} \tag{13}$$

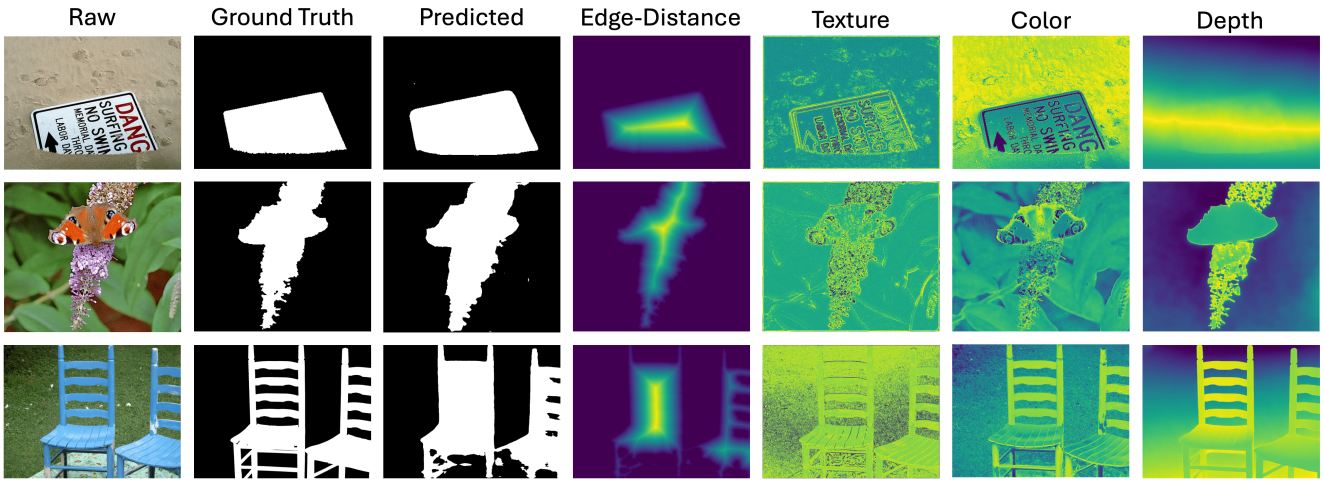

Figure 12. **C4. Diverse Image Characteristics.** Visualization of three scenes (warning sign, butterfly, chairs) and corresponding expert maps. Depending on the scene, different cues (edge distance, depth, color, and texture) can be highly informative, neutral, or misleading, illustrating why a single fixed prompting heuristic is insufficient.

- **Brightness Similarity.** Computes the mean luminance ($\bar{Y}$) over M_0 and scores pixels by how close they are to it, where pixels whose luminance is closer to $\bar{Y}$ are scored closer to 1.

$$\bar{Y} = \frac{1}{|M_0|} \sum_{u \in M_0} Y(x,u) \tag{14}$$

$$\mathcal{E}_{\text{bri}}(x,p) = 1 - \frac{\left|Y(x,p) - \bar{Y}\right|}{\max u \in M_0 \left|Y(x,u) - \bar{Y}\right|} \tag{15}$$

- **Texture Similarity.** Computes the mean gradient/texture response ($\bar{T}$) over M_0 and scores pixels by how close they are to it, where $T(x,p)$ is the Sobel gradient magnitude of the luminance.

$$T(x,p) = \sqrt{(\partial_x Y(x,p))^2 + (\partial_y Y(x,p))^2} \tag{16}$$

$$\bar{T} = \frac{1}{|M_0|} \sum_{u \in M_0} T(x,u) \tag{17}$$

$$\mathcal{E}_{\text{tex}}(x,p) = 1 - \frac{\left|T(x,p) - \bar{T}\right|}{\max u \in M_0 \left|T(x,u) - \bar{T}\right|} \tag{18}$$

- **Contrast Similarity.** Measures how much each pixel's local contrast matches the average contrast inside M_0. Let $\mu(x,p)$ be the 7×7 local mean around p, and $K(x,p)$ be the local contrast around p.

$$\mu(x,p) = \frac{1}{49} \sum_{u \in \mathcal{N}7\times7(p)} Y(x,u) \tag{19}$$

$$K(x,p) = \left|, Y(x,p) - \mu(x,p), \right| \tag{20}$$

$$\bar{K} = \frac{1}{|M_0|} \sum u \in M_0 K(x,u) \tag{21}$$

$$\mathcal{E}_{\text{ctr}}(x,p) = 1 - \frac{\left|K(x,p) - \bar{K}\right|}{\max u \in M_0 \left|K(x,u) - \bar{K}\right|} \tag{22}$$

- **Depth Similarity.** Computes the mean depth ($\bar{D}$) over M_0 and scores pixels by how close they are to it, where pixels whose depth is closer to $\bar{D}$ are scored higher.

$$\bar{D} = \frac{1}{|M_0|} \sum_{u \in M_0} D(x,u) \tag{23}$$

$$\mathcal{E}_{\text{depth}}(x,p) = 1 - \frac{\left|D(x,p) - \bar{D}\right|}{\max u \in M_0 \left|D(x,u) - \bar{D}\right|} \tag{24}$$

In the absence of ground-truth depth labels, we obtain $D(x,\cdot)$ from a monocular depth estimator. In our experiments we use *Marigold* [21] to predict per-pixel depth, but any comparable monocular depth model could be substituted.

F.3. Region Proposal Coverage

- **Superpixel Coverage.** Segments the image into regions (superpixels) and scores each pixel by the fraction of its region that is covered by M_0 (i.e., the proportion of pixels u in the same region as p that also belong to M_0).

$$\text{cov}(x,p) = \frac{\left|\{\, u \in M_0 : \text{R}(x,u) = \text{R}(x,p) \,\}\right|}{\left|\{\, u : \text{R}(x,u) = \text{R}(x,p) \,\}\right|} \tag{25}$$

$$\mathcal{E}_{\text{sp}}(x,p) = \text{cov}_{\text{sp}}(x,p). \tag{26}$$

In this work, we compute 200 SLIC superpixels using `scikit-image` [43], but any alternative superpixel method could be substituted, and the number of regions can be tailored to the target resolution or dataset.

- **SAM-Everything Object Coverage.** Reuses the coverage formulation in Eq. 25, but $\mathrm{R}(x, \cdot)$ now denotes the per-pixel region ID returned by SAM-Everything (i.e., SAM proposals) rather than SLIC superpixels:

$$\mathcal{E}_{\text{sam}}(x, p) = \text{cov}_{\text{sam}}(x, p). \tag{27}$$

- **SAM-Everything Weighted Object Coverage.** Similarly, reuses Eq. 25, but further emphasizes SAM proposals that are both highly contained in M_0 and account for a large share of M_0; thus very small but pure regions and very large but impure regions are downweighted, while regions that are simultaneously accurate and high-contribution to M_0 are favored.

$$w_{\text{sam}}(x, p) = \frac{\left|\{\, u \in M_0 : \mathrm{R}_{\text{sam}}(x, u) = \mathrm{R}_{\text{sam}}(x, p) \,\}\right|}{|M_0|}$$

$$\mathcal{E}_{\text{sam-w}}(x, p) = \text{cov}_{\text{sam}}(x, p) \cdot w_{\text{sam}}(x, p). \tag{28}$$

- **SAM Box Query.** Computes a tight bounding box around M_0, queries SAM with that box, and uses the returned mask as the expert score:

$$\mathcal{E}_{\text{sam-box}}(x, p) = M_{\text{sam-box}}(x, p), \tag{29}$$

where $M_{\text{sam-box}}(x, p) \in [0, 1]$ is the SAM mask predicted from the box around M_0.

F.4. Object Coverage vs. Weighted Coverage

While both SAM-Everything Object Coverage and Weighted Object Coverage are defined similarly, they target distinct failure cases. As shown in Fig. 13, SAM-Everything Object Coverage takes the initial mask and returns a closely aligned, high-confidence segmentation, whereas Weighted Object Coverage produces a similar mask but down-weights subcomponents such as the appendages in row one or the car door in row two. In row three, however, this weighting becomes advantageous: the weighted coverage correctly suppresses background leakage that base object coverage would otherwise include at high confidence.

F.5. Example Expert Maps

In Fig. 14, we provide an example of the ten expert maps computed for an image, highlighting their distinct signals. Edge-distance concentrates high scores near the center of the coarse dog mask and smoothly decays toward the boundary, suppressing most of the background. The color- and brightness-similarity maps share a similar overall structure, but brightness exhibits higher peaks over the sunlit fur on the dog's back and deeper troughs along the shaded legs and paws. Texture and contrast both emphasize high-frequency structure, highlighting the fur, leash, and ground detail. However, the texture map better captures edges and the contrast map provides a smoother output. Depth separates the foreground dog and pinecone from the distant background with a clear depth gradient, but struggles to separate the dog from the ground, where the true depth is similar. Finally, the three SAM-based experts (object coverage, weighted coverage, and box prompt) all provide a coarse reference to the dog, but differ in how they treat extremities such as the harness, pinecone, and leash.

Figure 13. Comparison of SAM-Everything Object Coverage vs. Weighted Object Coverage, highlighting applications on distinct failure cases.

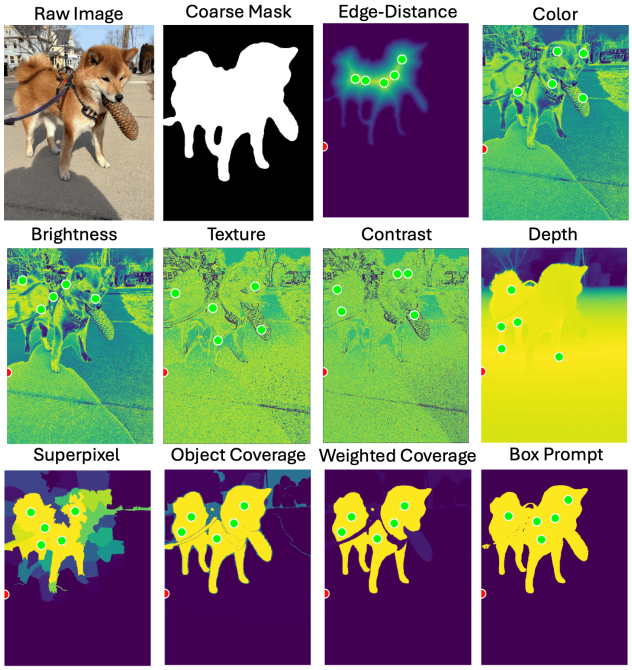

Figure 14. Example of Image-Informed Prompting (IIP) expert set for an image-mask pair, including Edge Distance, Color, Brightness, Texture, Contrast, Depth, Superpixel, Object Coverage, Weighted Object Coverage, and Box Prompt. Additionally, PPE-selected points from single-expert placement are included with positive point prompts in green, and the negative point prompt in red.

G. Adaptive PPE Radius Examples

In this work, we propose Prompt-Placement Explorer (PPE) to generate high-confidence, spatially-aware prompts. To achieve this, we implement a geometry-aware adaptive radius that grows and shrinks depending on the prompt location, ensuring that a consistent proportion of nearby mask pixels are suppressed while respecting the object's shape. For example, when placed at the center of the cow in Fig. 15, it has a relatively small suppressed region with a radius of 38 pixels. When placed along the perimeter, this radius increases to 60 to maintain an equal-sized suppressed region. When placed along the perimeter of a thin structure, this radius must grow even further up to 91 pixels, ensuring a stable suppression, regardless of the experts selected.

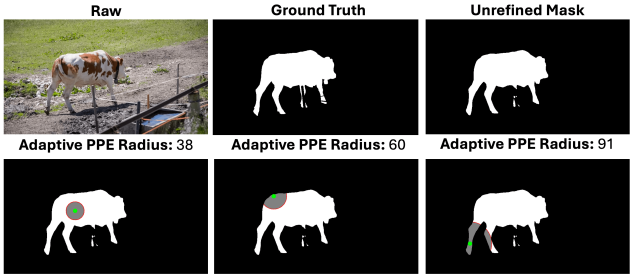

Figure 15. Example of adaptive PPE radius, highlighting spatially-aware prompting. With a $\lambda = 0.1$, our PPE seeks to suppress the nearest 10% of the mask. Depending on placement, this results in a radius of 38 pixels (left), 60 pixels (center), or 91 pixels (right).

H. Dataset Selection

In this work, we included five datasets, BIG, DAVIS585, ECSSD, MSRA-B, and VOC, specifically chosen for their relevance in segmentation refinement literature. For instance segmentation, DAVIS585 [7] was the primary dataset used for development and evaluation of SAMRefiner [29]. BIG was introduced by the authors of CascadePSP [9] as a high-resolution benchmark for refinement, served as the primary evaluation for SegRefiner [45], and was also evaluated by SAMRefiner [29]. Alternatively, VOC12-Val [14] is a more traditional vision benchmark, but was also evaluated by both CascadePSP [9] and SAMRefiner [29]. Finally, for salient object segmentation, we include ECSSD [40] and MSRA-B [20, 30], both used in CascadePSP's pipeline, offering diverse object scales with cluttered, textured backgrounds that stress boundary precision [9].

I. Statistical Significance

In this work, we evaluated statistical significance using non-parametric bootstrapping with 95% confidence intervals. For each SOTA method compared against, we computed $\Delta\Delta$IoU and $\Delta\Delta$BIoU, the average change in performance of PromptMoE over that method, as shown in Table 6 and Table 7. In this work we also provide PromptMoE-Lite, constraining our Dynamic Expert Selector (DES) to the six most efficient (non-learning-based) experts, offering reduced latency with comparable performance. Thus, we also provide the statistical significance of $\Delta\Delta$IoU and $\Delta\Delta$BIoU for PromptMoE-Lite in Table 8 and Table 9.

Table 5. Characteristics of Benchmark Datasets Evaluated.

	Dataset				
Metric	BIG	DAVIS585	ECSSD	MSRA-B	VOC
Images	100	300	1000	5000	1449
Vision Task	Semantic	Instance	Salient	Salient	Semantic
Landscape (%)	85.00%	100.00%	75.30%	70.92%	79.37%
Portrait (%)	15.00%	0.00%	22.70%	27.32%	19.67%
Square (%)	0.00%	0.00%	2.00%	1.76%	0.97%
Avg. Long Side	3639.89	865.80	400.00	399.62	496.44
Avg. Short Side	2525.36	480.00	286.56	293.17	359.60

For PromptMoE, we observed statistically significant improvement in IoU over CascadePSP-Fast, CascadePSP-Slow, SegRefiner-LR, DualSight, and SAMRefiner on all five datasets (Tab. 6). Compared against SegRefiner-HR, we observed statistically significant gains on four of the five datasets (DAVIS585, ECSSD, MSRA-B, and VOC). On BIG, SegRefiner-HR achieved a slightly higher mean ΔIoU, but with a $\Delta\Delta$IoU confidence interval $[-3.61, +1.51]$ crossing zero, this difference is not statistically significant. Evaluating BIoU, PromptMoE had statistically significant gains over CascadePSP-Fast, CascadePSP-Slow, SegRefiner-LR, and DualSight on all five datasets (Tab. 7). Compared to SAMRefiner, results were not statistically significant, but trended positive for DAVIS585 (CI: $[-0.18, +0.85]$) and MSRA-B (CI: $[-0.06, +0.30]$). However, when computing the statistical significance over all datasets, the mean $\Delta\Delta$BIoU of PromptMoE remains positive and statistically significant over both SAMRefiner (95% CI: $[+0.26, +1.19]$) and SegRefiner-HR (95% CI: $[+14.70, +18.15]$).

For PromptMoE-Lite, we observed very similar performance with an average ΔIoU/BIoU of +6.16/+8.93 compared to +6.24/+8.99 for PromptMoE (Tab. 20), and reduced runtime by 37.9% (Tab. 19). Despite restricting the learning-based experts, we still observed statistically significant IoU improvements over CascadePSP-Fast, CascadePSP-Slow, SegRefiner-LR, DualSight, and SAMRefiner on all five datasets as well (Tab. 8). Similar to PromptMoE, SegRefiner-HR had a higher average ΔIoU than PromptMoE-Lite on BIG (+9.55 vs. +8.54, Tab. 1), but this difference was not statistically significant. Evaluating BIoU, we saw similar trends with improvement over CascadePSP-Fast, CascadePSP-Slow, SegRefiner-LR, and DualSight on all five datasets, and statistically significant improvements over SAMRefiner on three of five datasets.

Table 6. Head-to-head 95% confidence intervals for ΔΔIoU (percentage points), comparing *PromptMoE* against each comparator (row) across datasets (columns). ΔΔ = ΔOurs − ΔComparator.

Comparator	BIG	DAVIS585	ECSSD	MSRA-B	VOC	Mean ΔΔIoU
CascadePSP-Fast [9]	[+2.12, +6.90]	[+5.57, +14.21]	[+4.64, +8.68]	[+4.16, +7.22]	[+6.15, +8.71]	[+4.53, +9.15]
CascadePSP-Slow [9]	[+1.16, +6.08]	[+2.01, +8.04]	[+3.77, +6.79]	[+3.00, +4.92]	[+4.44, +7.68]	[+2.87, +6.70]
SegRefiner-LR [45]	[+3.41, +8.14]	[+15.93, +22.03]	[+19.11, +20.52]	[+15.50, +16.06]	[+11.13, +12.34]	[+13.02, +15.82]
SegRefiner-HR [45]	[-3.61, +1.51]	[+11.86, +17.14]	[+20.08, +21.88]	[+15.13, +16.49]	[+11.11, +12.59]	[+10.91, +13.92]
DualSight [35]	[+3.05, +6.32]	[+0.60, +3.94]	[+3.00, +5.20]	[+2.33, +3.60]	[+4.07, +5.32]	[+2.61, +4.88]
SAMRefiner [29]	[+0.38, +2.94]	[+0.06, +0.59]	[+0.58, +1.20]	[+0.27, +0.62]	[+0.59, +1.17]	[+0.38, +1.30]

Table 7. Head-to-head 95% confidence intervals for ΔΔBIoU (percentage points), comparing *PromptMoE* against each comparator (row) across datasets (columns). ΔΔ = ΔOurs − ΔComparator.

Comparator	BIG	DAVIS585	ECSSD	MSRA-B	VOC	Mean ΔΔBIoU
CascadePSP-Fast [9]	[+4.61, +9.26]	[+4.72, +14.14]	[+7.23, +15.04]	[+7.62, +13.73]	[+10.27, +13.67]	[+6.89, +13.17]
CascadePSP-Slow [9]	[+2.57, +6.91]	[+1.09, +6.77]	[+6.80, +12.02]	[+5.61, +9.02]	[+8.15, +11.68]	[+4.84, +9.28]
SegRefiner-LR [45]	[+6.21, +11.27]	[+18.03, +24.46]	[+31.62, +33.29]	[+27.40, +28.02]	[+15.67, +17.38]	[+19.79, +22.89]
SegRefiner-HR [45]	[-4.01, +1.13]	[+8.28, +14.74]	[+30.84, +33.10]	[+25.98, +27.20]	[+12.42, +14.57]	[+14.70, +18.15]
DualSight [35]	[+4.91, +7.92]	[+0.04, +6.04]	[+5.08, +7.64]	[+2.70, +4.39]	[+3.33, +5.19]	[+3.21, +6.24]
SAMRefiner [29]	[+0.53, +2.46]	[-0.18, +0.85]	[+0.63, +1.34]	[-0.06, +0.30]	[+0.38, +0.97]	[+0.26, +1.19]

Table 8. Head-to-head 95% confidence intervals for ΔΔIoU (percentage points), comparing **PromptMoE-Lite** against each comparator (row) across datasets (columns). ΔΔ = ΔOurs − ΔComparator.

Comparator	BIG	DAVIS585	ECSSD	MSRA-B	VOC	Mean
CascadePSP-Fast [9]	[+2.01, +6.32]	[+5.62, +13.99]	[+4.59, +8.63]	[+4.16, +7.29]	[+6.10, +8.67]	[+4.49, +8.98]
CascadePSP-Slow [9]	[+1.05, +5.61]	[+2.04, +7.82]	[+3.72, +6.74]	[+2.98, +4.97]	[+4.39, +7.65]	[+2.84, +6.56]
SegRefiner-LR [45]	[+3.28, +7.62]	[+16.00, +21.81]	[+19.09, +20.49]	[+15.53, +16.09]	[+11.10, +12.30]	[+13.00, +15.66]
SegRefiner-HR [45]	[-3.80, +0.98]	[+11.91, +16.94]	[+20.10, +21.84]	[+15.20, +16.48]	[+11.08, +12.56]	[+10.90, +13.76]
DualSight [35]	[+2.78, +6.01]	[+0.65, +3.72]	[+3.01, +5.17]	[+2.31, +3.63]	[+4.05, +5.27]	[+2.56, +4.76]
SAMRefiner [29]	[+0.03, +2.67]	[+0.02, +0.43]	[+0.58, +1.16]	[+0.31, +0.64]	[+0.56, +1.13]	[+0.30, +1.21]

Table 9. Head-to-head 95% confidence intervals for ΔΔBIoU (percentage points), comparing **PromptMoE-Lite** against each comparator (row) across datasets (columns). ΔΔ = ΔOurs − ΔComparator.

Comparator	BIG	DAVIS585	ECSSD	MSRA-B	VOC	Mean
CascadePSP-Fast [9]	[+4.53, +8.94]	[+4.72, +14.04]	[+7.26, +14.94]	[+7.66, +13.77]	[+10.18, +13.65]	[+6.87, +13.07]
CascadePSP-Slow [9]	[+2.50, +6.70]	[+1.10, +6.70]	[+6.83, +11.95]	[+5.63, +9.08]	[+8.02, +11.42]	[+4.82, +9.17]
SegRefiner-LR [45]	[+6.03, +11.01]	[+18.11, +24.41]	[+31.62, +33.24]	[+27.43, +28.07]	[+15.59, +17.31]	[+19.76, +22.81]
SegRefiner-HR [45]	[-4.14, +0.78]	[+8.29, +14.65]	[+30.85, +33.05]	[+26.06, +27.20]	[+12.33, +14.49]	[+14.68, +18.03]
DualSight [35]	[+4.78, +7.83]	[+0.03, +6.02]	[+5.09, +7.48]	[+2.72, +4.43]	[+3.23, +5.14]	[+3.17, +6.18]
SAMRefiner [29]	[+0.33, +2.39]	[-0.18, +0.81]	[+0.58, +1.31]	[-0.01, +0.32]	[+0.30, +0.87]	[+0.20, +1.14]

J. Ablation Analysis

To evaluate the improvement from our proposed strategies, we performed an ablation study, systematically removing PromptMoE components to measure their impact. While Tab. 2 (Sec. 4.3) reports the mean ΔIoU/BIoU over all five datasets, we also report per-dataset scores, as shown in Tab. 10. Analyzing these results, using a single point prompt (1 PP) severely degraded mask performance, lowering ΔIoU/BIoU by an average of -18.82/15.14 on all five datasets. Including a bounding box (1 PP + B) significant improved performance, resulting in an average ΔBIoU gain of +3.62, and decreasing the ΔIoU loss from -18.82 to -0.71, and even improving performance on DAVIS585 (+0.13) and VOC (+1.26). Introducing a coarse mask (1 PP + B + M) resulted in the largest increase in performance, producing average gains of +4.71/+5.48 and consistent IoU/BIoU gains across all five datasets. With a negative point (1 PP + 1 NP + B + M), we make a SAMRefiner-equivalent style prompt, yielding additional

Table 10. Quantitative evaluation of *PromptMoE* ablation, highlighting the individual impact of Dynamic Expert Selection (DES), Prompt-Placement Exploration (PPE), the addition of multiple inclusion points, and overall prompt composition. We report the mean improvement (Δ) in Intersection-over-Union (IoU) and Boundary IoU relative to the unrefined base masks for each dataset. Positive values indicate improvement, negative indicate degradation. The rightmost column reports a macro-average, equally weighting all five datasets.

Method	BIG	DAVIS585	ECSSD	VOC	MSRA-B	Mean ΔIoU / ΔBIoU
Unrefined	78.25 / 70.11	80.05 / 83.00	81.41 / 70.23	66.73 / 60.08	75.15 / 61.88	76.32 / 69.06
1 PP	−20.87 / −17.45	−19.21 / −20.66	−22.02 / −16.62	−13.01 / −9.92	−18.98 / −11.04	−18.82 / −15.14
1 PP + B	−0.82 / +3.36	+0.13 / −0.74	−2.37 / +3.18	+1.26 / +5.68	−1.75 / +6.63	−0.71 / +3.62
1 PP + B + M	+5.71 / +8.58	+3.30 / +2.06	+3.91 / +8.79	+6.69 / +9.31	+3.94 / +10.31	+4.71 / +7.81
1 PP + 1 NP + B + M	+7.04 / +9.87	+3.48 / +2.10	+4.86 / +9.68	+7.33 / +9.89	+4.66 / **+10.53**	+5.48 / +8.42
5 PP + 1 NP + B + M	+6.93 / +9.94	+3.51 / +2.08	+5.50 / +10.24	+7.31 / +9.82	+4.89 / +10.44	+5.63 / +8.50
5 PP + 1 NP + B + M + PPE	+7.92 / +10.82	+3.50 / +2.09	+5.77 / +10.51	+7.74 / +10.21	+5.07 / +10.51	+6.00 / +8.83
5 PP + 1 NP + B + M + PPE + DES	**+8.54** / **+11.01**	**+3.64** / **+2.35**	**+5.99** / **+10.67**	**+7.94** / **+10.43**	**+5.10** / +10.48	**+6.24** / **+8.99**

gains to +5.48/+8.42.

Improving over this SAMRefiner-style baseline, we next introduce multiple inclusion points under the same prompt composition. However, naively adding four extra positives without PPE (5 PP + 1 NP + B + M) yields only marginal gains, increasing the mean ΔIoU/BIoU from +5.48/+8.42 to +5.63/+8.50, and even slightly worsening performance on BIG and VOC. In contrast, when these additional points are combined with PPE (5 PP + 1 NP + B + M + PPE), enforcing spatially diversity and shape-aware prompting, they consistently improve performance on all datasets, pushing the average improvement to +6.00/+8.83. Finally, enabling DES on top of this configuration (full *PromptMoE*, 5 PP + 1 NP + B + M + PPE + DES) yields an average ΔIoU/BIoU of +6.24/+8.99 and achieves the highest improvements on all five datasets in this ablation.

K. Dynamic Expert Selector (DES) Evaluation

To benchmark our DES, we compared it against a series of alternative expert-selection strategies, reporting their average ΔIoU/BIoU over the unrefined masks on five datasets, as shown in Tab. 11. This includes worst-case oracle (lower bound), best single-expert oracle, average single-expert (mean over all ten experts), random single-expert (mean over five randomly sampled single experts), and a dense mixture (all ten experts activated). Additionally, we compare DES with an activation budget of one, two, three, and four experts. Here, our worst-case oracle yielded a mean gain of +3.47/+5.95, highlighting that even the worst single-expert for a given image-mask pair consistently improved performance on all five datasets, confirming a well-selected set of experts. Comparing non-oracle strategies, random single-expert and average single-expert performed similarly (+5.91/+8.71 vs. +5.90/+8.71) with little variation between datasets, while dense evaluation performed slightly better at +6.01/+8.80. Comparing our DES router, using an activation budget of two experts performed the best on average (+6.24/+8.99), outscoring single-expert (+5.90/+8.80), three-expert (+5.92/+8.68) and four-expert (+5.87/+8.66) solutions. Overall, these results show that DES with two active experts per image balances additional complementary cues while avoiding signal dilution.

L. Prompt-Placement Explorer Evaluation

Additionally, in this work, we studied how the number of inclusion points k and the separation factor λ affect PPE performance, as shown in Fig. 16. Specifically, we varied λ from 0 to 0.1 (steps at $\lambda = 0.02$) and k from 1 to 9 (steps at $k = 2$), and found that an increase in λ and k consistently improved performance, to a point. For example, moving from one to three points raises mean IoU from +5.45 to +5.86 and raising λ from 0 to 0.04 pushes it to +6.01. However, increasing λ or k too far eventually degraded performance. For points, increasing from $k = 5$ to $k = 7$ consistently resulted in a decrease in performance across all λ values for both ΔIoU and ΔBIoU. For suppression factor, the threshold was partially dependent on k. For example, the maximum ΔIoU was scored with a λ of 0.04 at $k = 5$, 0.06 at $k = 7$, and 0.08 at $k = 9$. While these results demonstrate that PromptMoE is sensitive to parameter selection, even when λ or k was higher than optimal, PromptMoE still consistently performed better than when $k = 1$ or $\lambda = 0$ (indicating no PPE).

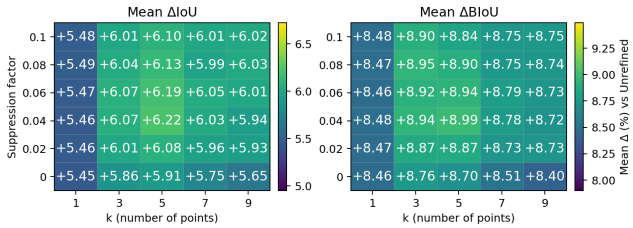

Figure 16. Quantitative evaluation of PPE parameters (Seperation Factor λ and number of points k) on performance. We report mean ΔIoU/BIoU over five benchmark datasets (BIG, DAVIS585, ECSSD, MSRA-B, and VOC).

Table 11. Unrefined vs. oracle/average/random/all/routers across datasets. Unrefined shows raw IoU/BIoU means (%). All other rows report mean ΔIoU / ΔBIoU (pp, signed), equal-weight over base models per dataset. Bolding ignores the Best-oracle row.

Method	BIG	DAVIS585	ECSSD	VOC	MSRA-B	Mean ΔIoU / ΔBIoU
Unrefined	78.25 / 70.11	80.05 / 83.00	81.41 / 70.23	66.73 / 60.08	75.15 / 61.88	76.32 / 69.06
Worst single (oracle)	+4.78 / +7.39	+2.04 / +0.44	+3.31 / +7.55	+4.25 / +6.41	+2.98 / +7.96	+3.47 / +5.95
Random (avg×5)	+7.57 / +10.34	+3.59 / +2.24	+5.78 / +10.41	+7.58 / +10.12	+5.02 / +10.39	+5.91 / +8.70
Average single	+7.56 / +10.38	+3.55 / +2.27	+5.76 / +10.40	+7.59 / +10.11	+5.03 / +10.40	+5.90 / +8.71
All experts	+7.58 / +10.48	+3.66 / +2.18	+5.91 / +10.57	+7.67 / +10.15	**+5.22 / +10.61**	+6.01 / +8.80
Router (single)	+8.17 / +10.92	+3.57 / +2.29	+5.56 / +10.37	+7.38 / +10.05	+4.81 / +10.38	+5.90 / +8.80
Router (pair)	**+8.54 / +11.01**	+3.64 / +2.35	**+5.99 / +10.67**	**+7.94 / +10.43**	+5.10 / +10.48	**+6.24 / +8.99**
Router (Triplet)	+7.19 / +9.86	+3.66 / +2.33	+5.78 / +10.39	+7.84 / +10.31	+5.16 / +10.50	+5.92 / +8.68
Router (Quadruplet)	+6.90 / +9.67	**+3.77 / +2.52**	+5.83 / +10.43	+7.67 / +10.17	+5.15 / +10.52	+5.87 / +8.66
Best single (oracle)	+9.91 / +12.95	+4.95 / +4.00	+7.46 / +12.66	+10.56 / +13.51	+6.62 / +12.55	+7.90 / +11.13

Table 12. Stratified evaluation of **PromptMoE**, reporting ΔIoU/BIoU refinement performance across different initial IoU bands.

Initial IoU bin	BIG	DAVIS585	ECSSD	VOC	MSRA-B	Mean
0-20	+12.01 / +7.84	+10.62 / +5.46	+1.33 / +8.30	+9.62 / +5.81	-0.77 / +5.43	+6.56 / +6.57
20-40	+29.57 / +17.51	+14.16 / -0.65	+11.70 / +14.54	+14.75 / +11.83	+3.25 / +8.45	+14.69 / +10.33
40-60	+13.52 / +11.93	-10.62 / -12.09	+15.13 / +21.02	+14.55 / +13.03	+10.29 / +14.66	+8.57 / +9.71
60-80	+16.71 / +22.62	+5.78 / +3.56	+11.91 / +18.69	+8.13 / +11.20	+10.60 / +17.38	+10.63 / +14.69
80-100	+3.08 / +7.24	+2.73 / +2.07	+4.00 / +8.03	+3.11 / +9.46	+3.73 / +8.87	+3.33 / +7.14

M. Stratified Evaluation of Performance Improvement

PromptMoE's performance is dependent on the initial unrefined input mask. A stratified evaluation (Table 12) revealed mean IoU improvements of +6.56 for masks with initial IoU 0-20, +14.69 for 20-40, +8.57 for 40-60, +10.63 for 60-80, and +3.33 for 80-100. While *PromptMoE* successfully raises IoU in each strata, this highlights that final performance remains tied to initial quality. For example, despite large gains in the 20-40 range, many results would still remain below 50 IoU. To achieve segmentations with an IoU near 100, the unrefined segmentation would likely need to be closer.

N. Evaluating Backbone Portability

We conducted an ablation of backbone architectures, evaluating SAM's three trained ViT backbones [13, 25]: ViT-B, ViT-L, and ViT-H. While our ultimate solution uses the largest backbone, ViT-H, we explored utilization of the two smaller architectures to evaluate portability. As shown in Tab. 13, performance scales with backbone size: ViT-H → ViT-L drops from +6.24/+8.99 improvement over initial segmentations to +5.74/+7.90, and ViT-L → ViT-B drops further to +3.99/+5.46. However, crucially, *PromptMoE* can be used on all three SAM backbones and still delivers high-quality refined segmentations.

Table 13. Quantitative evaluation of Vision Transformer (ViT) backbone impact on performance .

Backbone	Mean ΔIoU	Mean ΔBIoU
ViT-B	+3.99	+5.46
ViT-L	+5.74	+7.90
ViT-H	**+6.24**	**+8.99**

N.1. PromptMoE with SAM-High Quality

In addition to comparing PromptMoE with varying backbone capacity (Tab. 13), we also explored the effect of different architectures. Thus, we explored PromptMoE's compatibility with SAM-HQ [22], a more advanced mask-decoder for improved final segmentations. As shown in Tab. 14, PromptMoE's prompt generation framework is compatible, and even higher performance, with SAM-HQ, underscoring the generalizability of PromptMoE. Across all five datasets, PromptMoE with SAM-HQ scored the *highest*, with an average IoU/BIoU of +7.20/+9.74. Per-dataset, PromptMoE-HQ scored the highest IoU against all non-HQ methods on *every* dataset, even outscoring SegRefiner-HR on the BIG dataset, which SegRefiner-HR was explicitly trained for. Compared to SAMRefiner-HQ under the same advanced SAM-HQ architecture, PromptMoE-HQ scored higher on four of five datasets, with average ΔIoU/ΔBIoU gains of +7.20/+9.74 compared to +6.83/+9.25. Addition-

Table 14. Comparative study with state-of-the-art refinement methods across 5 benchmark datasets. Metrics are mean improvement (Δ) in Intersection-over-Union (IoU) and Boundary IoU over the unrefined base masks. Positive values indicate improvement and negative indicate degradation of mask quality. Our method is highlighted in green.

Method	BIG	DAVIS585	ECSSD	MSRA-B	VOC	Mean ΔIoU / ΔBIoU
Unrefined	78.25 / 70.11	80.05 / 83.00	81.41 / 70.23	75.15 / 61.88	66.73 / 60.08	76.32 / 69.06
CascadePSP-Fast [9]	+4.29 / +4.18	−6.21 / −6.99	−0.99 / −1.21	−1.01 / −1.01	+0.42 / −1.30	−0.70 / −1.27
CascadePSP-Slow [9]	+4.97 / +6.27	−1.29 / −1.47	+0.62 / +0.98	+0.93 / +2.71	+1.71 / +0.73	+1.39 / +1.85
SegRefiner-LR [45]	+2.96 / +2.35	−15.34 / −18.92	−13.81 / −21.77	−10.68 / −17.25	−3.82 / −6.16	−8.14 / −12.35
SegRefiner-HR [45]	+9.55 / +12.51	−10.85 / −9.10	−15.01 / −21.37	−10.67 / −16.16	−3.86 / −3.05	−6.17 / −7.43
DualSight [35]	+3.89 / +4.63	+1.48 / −0.57	+1.99 / +4.50	+2.08 / +6.79	+3.29 / +6.29	+2.55 / +4.33
SAMRefiner [29]	+6.84 / +9.50	+3.33 / +2.03	+5.10 / +9.69	+4.67 / +10.35	+7.05 / +9.74	+5.40 / +8.26
SamRefiner-HQ [29]	+9.88 / +12.37	+3.70 / +2.27	+6.76 / +11.43	**+5.76** / +10.51	+8.04 / +9.68	+6.83 / +9.25
PromptMoE (Ours)	+8.54 / +11.01	+3.64 / +2.35	+5.99 / +10.67	+5.10 / +10.47	+7.94 / **+10.43**	+6.24 / +8.99
PromptMoE-HQ (Ours)	**+10.65 / +13.21**	**+3.89 / +2.61**	**+6.94 / +11.89**	+5.73 / **+10.61**	**+8.78** / +10.40	**+7.20 / +9.74**

Table 15. Head-to-head 95% confidence intervals for ΔΔIoU (percentage points), comparing **PromptMoE-HQ** against each comparator (row) across datasets (columns). ΔΔ = ΔOurs − ΔComparator.

Comparator	BIG	DAVIS585	ECSSD	MSRA-B	VOC	Mean
CascadePSP-Fast [9]	[+3.72, +8.96]	[+5.76, +14.58]	[+5.14, +9.72]	[+4.48, +8.08]	[+6.65, +9.50]	[+5.15, +10.17]
CascadePSP-Slow [9]	[+2.70, +8.49]	[+2.16, +8.39]	[+4.26, +7.82]	[+3.30, +5.76]	[+4.94, +8.46]	[+3.47, +7.79]
SegRefiner-LR [45]	[+5.55, +10.15]	[+16.10, +22.39]	[+20.01, +21.50]	[+16.02, +16.81]	[+12.07, +13.10]	[+13.95, +16.79]
SegRefiner-HR [45]	[-0.95, +3.41]	[+12.01, +17.52]	[+21.29, +22.62]	[+15.98, +16.85]	[+12.09, +13.30]	[+12.08, +14.74]
DualSight [35]	[+5.18, +8.42]	[+0.73, +4.31]	[+3.99, +6.23]	[+2.64, +4.33]	[+4.91, +6.12]	[+3.49, +5.88]
SamRefiner [29]	[+2.68, +5.06]	[+0.16, +1.01]	[+1.46, +2.31]	[+0.70, +1.36]	[+1.40, +2.09]	[+1.28, +2.37]
SamRefiner-HQ [29]	[+0.25, +1.61]	[+0.03, +0.37]	[+0.05, +0.36]	[-0.12, +0.06]	[+0.49, +0.99]	[+0.14, +0.68]
PromptMoE	[+1.06, +3.54]	[-0.03, +0.54]	[+0.50, +1.40]	[+0.32, +0.94]	[+0.46, +1.31]	[+0.46, +1.55]

Table 16. Head-to-head 95% confidence intervals for ΔΔBIoU (percentage points), comparing **PromptMoE-HQ** against each comparator (row) across datasets (columns). ΔΔ = ΔOurs − ΔComparator.

Comparator	BIG	DAVIS585	ECSSD	MSRA-B	VOC	Mean
CascadePSP-Fast [9]	[+6.35, +11.65]	[+5.21, +14.20]	[+8.53, +15.95]	[+7.91, +13.90]	[+10.17, +13.64]	[+7.63, +13.87]
CascadePSP-Slow [9]	[+4.09, +9.61]	[+1.60, +6.90]	[+8.03, +12.94]	[+5.85, +9.16]	[+7.97, +11.50]	[+5.51, +10.02]
SegRefiner-LR [45]	[+8.45, +13.33]	[+18.58, +24.57]	[+33.03, +34.32]	[+27.57, +28.17]	[+15.84, +17.24]	[+20.69, +23.53]
SegRefiner-HR [45]	[-1.51, +3.12]	[+8.80, +14.79]	[+32.26, +34.17]	[+26.15, +27.35]	[+12.57, +14.46]	[+15.65, +18.78]
DualSight [35]	[+6.89, +10.33]	[+0.54, +6.11]	[+6.53, +8.50]	[+2.94, +4.42]	[+3.29, +5.07]	[+4.04, +6.89]
SamRefiner [29]	[+2.55, +4.95]	[+0.25, +0.96]	[+1.66, +2.70]	[+0.08, +0.44]	[+0.25, +1.03]	[+0.96, +2.02]
SamRefiner-HQ [29]	[+0.34, +1.35]	[+0.02, +0.73]	[+0.28, +0.70]	[-0.01, +0.23]	[+0.41, +1.00]	[+0.21, +0.80]
PromptMoE	[+1.06, +3.63]	[-0.11, +0.60]	[+0.77, +1.62]	[-0.06, +0.31]	[-0.31, +0.26]	[+0.27, +1.29]

ally, PromptMoE-HQ was found to have statistically significant gains over nearly all methods for IoU *and* BIoU on *all five datasets*, except for SegRefiner-HR on BIG and SAMRefiner-HQ on MSRA-B, as shown in Tab. 15 and 16. For BIG, while not statistically significant, PromptMoE-HQ had a trending positive confidence interval of [-0.95,+3.41), and against SAMRefiner-HQ on MSRA-B, the confidence intervals remained narrowly overlapping, such as an BIoU interval of [-0.01,+0.23].

O. Comparison to SAM2 & SAM3

After the original release of SAM, SAM2 extended instance segmentation from images to video inputs [37], while the recently released SAM3 offers a unified prompting framework across multiple modalities [3]. To isolate the contributions of PromptMoE from the underlying SAM architecture, we performed an evaluation similar to Tab. 10, testing the refinement capabilities of SAM2 and SAM3 under three prompting settings: a single point, a point with a bounding box, and a point with a bounding box and a coarse mask. Across all settings, PromptMoE remained consistently superior, achieving a mean ΔIoU/BIoU of +6.24/+8.99, compared to SAM2 (-19.31/-12.83, -1.51/+2.65, +3.90/+6.09) and SAM3 (-17.45/-11.68, -1.21/+2.14, +2.29/+4.02). These results suggest that the primary bottleneck in refinement is prompt quality rather than raw model capacity,

further validating the need for targeted prompt-generation frameworks.

P. Latency Evaluation

To quantify the computational overhead of our method, we measure the average wall-clock time required to refine a single annotation for each method and dataset, summarized in Tab. 17. All timings were obtained on a single NVIDIA L40S GPU node with Intel Xeon Gold 6442Y CPUs, CUDA 12.7, and no competing jobs.

As a result of the innovations in PromptMoE, including image-mask encoding, routing, expert computation, and our Prompt-Placement Explorer (PPE), PromptMoE was more computationally expensive than previous refiners. To mitigate this, we investigate two complementary strategies: a PromptMoE-Lite configuration that restricts DES to a subset of efficient experts, substantially reducing latency while preserving most of the accuracy gains (Tables 19 and 20), and smaller SAM backbones (ViT-B/L) that further lower runtime while still allowing competitive segmentation refinement performance (Tab. 13), as will be discussed further in Supp. Sec. P.1.

Table 17. Average time (in seconds) to refine an annotation for each dataset from CascadePSP-Fast (PSP-F), CascadePSP-Slow (PSP-S), SegRefiner-LR (LR), SegRefiner-HR (HR), DualSight (DS), SAMRefiner (SR), and PromptMoE (Ours).

	Method						
Dataset	PSP-F	PSP-S	LR	HR	DS	SR	Ours
BIG	0.96	5.67	2.53	2.30	1.36	7.43	10.80
DAVIS585	0.16	0.18	0.37	0.32	0.30	0.59	1.07
ECSSD	0.13	0.15	0.15	0.15	0.19	0.22	0.44
MSRA-B	0.19	0.23	0.23	0.23	0.28	0.31	0.73
VOC	0.19	0.23	0.27	0.26	0.29	0.34	0.78

In addition to these PromptMoE variants, we also analyze how the total runtime of *PromptMoE* is split across its main components. We separate refinement time into five stages: *SAM Encoding* (image embedding), *DES* (mask/context encoding and routing), *Expert Maps* (computing the selected experts), *Prompt Generation & Encoding* (PPE, tight-box extraction, mask softening, and SAM prompt encoding), and *Mask Decoding* (final SAM prediction), as reported in Tab. 18. On four of the five datasets (DAVIS585, ECSSD, MSRA-B, and VOC), the *Expert Maps* stage is the most time-consuming component, followed closely by the initial *SAM Encoding*, while the remaining stages contribute comparatively little overhead. In particular, *DES* and *Prompt Preparation* are relatively lightweight on these benchmarks, requiring only $\sim$0.03 s and $\sim$0.09 s per refinement, on average. On the high-resolution BIG dataset, however, *Prompt Preparation* dominates the runtime (8.39 s). However, this bottleneck is not specific to just our PPE module. *Prompt Preparation* also includes bounding box extraction and mask processing, which scale in complexity with image resolution. A similar increase in refinement time is observed by SAMRefiner, increasing from an average of 0.37 seconds to 7.43 seconds. Future works will explore the impact of reduced-resolution inference with upsampling to further improve the speed-accuracy trade-off.

Table 18. Average wall-clock time (in seconds) for each of the five main components of the *PromptMoE* pipeline, broken down by dataset.

	Method				
Dataset	BIG	DAVIS585	ECSSD	MSRA-B	VOC
SAM Encoding	0.53	0.28	0.28	0.28	0.28
DES	0.08	0.04	0.02	0.02	0.02
Expert Maps	1.14	0.59	0.33	0.37	0.38
Prompt Prep	8.39	0.15	0.06	0.06	0.07
Mask Decoding	0.03	0.03	0.03	0.03	0.03

P.1. PromptMoE Variants

While *PromptMoE* is slower than SAMRefiner, Tab. 19 shows that our DES module substantially reduces the overhead relative to a dense expert mixture. Across all five benchmarks, PromptMoE is consistently faster than PromptMoE-Dense (on average reducing per-annotation runtime by $\sim$60%), while also slightly improving mean ΔIoU/ΔBIoU (+6.24/+8.99 vs. +6.01/+8.82 in Tab. 20).

Table 19. Average time (in seconds) to refine an annotation for each dataset from SAMRefiner (SR) and three variations of our proposed method, *PromptMoE*, including base (pairwise expert fusion from set of 10 experts), dense (all 10 experts activated for all 10 images), and lite (pairwise expert fusion from a subset of 6 experts)

	Method			
Dataset	SAMRefiner	PromptMoE	PromptMoE-Dense	PromptMoE-Lite
BIG	7.43	10.80	15.10	9.78
DAVIS585	0.59	1.07	2.63	0.52
ECSSD	0.22	0.44	1.60	0.27
MSRA-B	0.31	0.73	2.40	0.40
VOC	0.34	0.78	2.55	0.43

To give practitioners more control over quality-runtime trade-offs, we propose a PromptMoE-Lite variant that restricts DES to the six most efficient experts (excluding depth estimation, SAM object coverage, SAM weighted object coverage, and the SAM box expert). As shown in Tab. 20, PromptMoE-Lite retains most of PromptMoE's performance (+6.16/+8.93 vs. +6.24/+8.99) while reducing runtime by an average of $\sim$38% (Tab. 19). Compared to

SAMRefiner, PromptMoE-Lite offers statistically significant IoU gains on all five datasets (Tab. 8) while requiring less than a 20% increase in runtime on average.

Table 20. Perforamnce Comparison of SAMRefiner, PromptMoE, PromptMoE-Dense, and PromptMoE-Lite on five benchmarks (BIG, DAVIS585, ECSSD, MSRA-B, and VOC), reporting IoU and BIoU.

	Method			
Dataset	SAMRefiner	PromptMoE	PromptMoE-Dense	PromptMoE-Lite
BIG	+6.84 / +9.50	+8.54 / +11.01	+7.57 / +10.53	+8.23 / +10.86
DAVIS585	+3.33 / +2.03	+3.64 / **+2.35**	**+3.66** / +2.20	+3.55 / +2.32
ECSSD	+5.10 / +9.69	**+5.99 / +10.67**	+5.95 / +10.61	+5.97 / +10.64
MSRA-B	+4.67 / +10.35	+5.10 / +10.47	**+5.23 / +10.61**	+5.13 / +10.50
VOC	+7.05 / +9.74	**+7.94 / +10.43**	+7.66 / +10.14	+7.90 / +10.34
Mean	+5.40 / +8.26	**+6.24 / +8.99**	+6.01 / +8.82	+6.16 / +8.93

Finally, we evaluate the impact of scaling the SAM backbone from ViT-H to ViT-L and ViT-B. On the four non-BIG datasets (DAVIS585, ECSSD, MSRA-B, and VOC), moving from ViT-H to ViT-L and ViT-B reduces IoU gains by roughly 8.0% and 36.1%, respectively (Tab. 13, while the average runtime decreased by 34.4% (ViT-L) and 66.2% (ViT-B) relative to ViT-H (Tab. 21). These results highlight how PromptMoE can be tailored to a specific task depending on performance and latency constraints.

Table 21. Average time (in seconds) to refine an annotation for each dataset from PromptMoE at varying ViT backbones.

	Method		
Dataset	PromptMoE (ViT-H)	PromptMoE (ViT-L)	PromptMoE (ViT-B)
BIG	10.80	9.42	9.44
DAVIS585	1.07	0.60	0.40
ECSSD	0.44	0.41	0.186
MSRA-B	0.73	0.46	0.18
VOC	0.78	0.39	0.24

Q. Expert Selection Frequency

To further motivate Challenge 4 (Sec. 1), we computed how often each of our ten experts was the *best* single expert on each image-mask pair across the five datasets, summarized in Tab. 22. On average, *Texture* was the least frequently optimal (7.31%), whereas *SAM-Everything Object Coverage* and *SAM Box Query* were the most frequently optimal (14.01% and 14.04%, respectively). This optimality was also dataset-dependent. For example, *Edge Distance* was the top expert 20.28% of the time on VOC but only 6.96% on DAVIS585. This wide spread of *best* experts within a single dataset and large variation across datasets underscores the value of PromptMoE's dynamic routing. instead of committing to a single heuristic, it can activate the subset of experts that is most appropriate for the current image and coarse mask.

Table 22. Single-expert win rates for PromptMoE: percentage of images on which each expert achieves the highest IoU (ties split evenly), averaged equally across base models.

	Dataset					
Expert	BIG	DAVIS585	ECSSD	VOC	MSRA-B	Mean
Edge-Distance	9.55%	7.21%	9.75%	**20.11%**	**14.42%**	12.21%
Depth	6.69%	10.21%	10.65%	10.58%	12.12%	10.05%
Color	3.60%	8.20%	8.49%	8.50%	10.18%	7.79%
Superpixels	3.58%	7.38%	7.52%	12.15%	9.21%	7.97%
Contrast	9.20%	6.74%	7.93%	9.15%	8.67%	8.34%
Brightness	8.20%	6.87%	7.63%	9.29%	9.08%	8.21%
Texture	5.60%	6.19%	7.07%	9.53%	8.87%	7.45%
SAM Object Coverage	17.39%	16.96%	11.91%	6.32%	11.08%	12.73%
SAM Weighted Coverage	14.30%	10.55%	7.39%	6.00%	7.68%	9.18%
SAM Box Prompt	**21.89%**	**19.69%**	**21.67%**	8.37%	8.70%	**16.06%**

Evaluating the expert-selection frequency in Tab. 23, our DES router draws from *all* experts, with the least frequently selected being SAM-Everything Weighted Coverage at 6.86% on average, indicating no collapse to a subset of high-performing experts. Additionally, Tab. 23 highlights an adaptive distribution between datasets. For instance, Depth is activated 25.13% of the time on BIG but only 3.43% on ECSSD and Brightness jumps from 13.31% on MSRA-B to 52.73% on BIG. This variability confirms the "dynamic" part of DES: it adapts its active experts to the image/mask statistics of each dataset rather than applying a single global routing policy.

Table 23. Expert activation frequency (non-zero weights) for PromptMoE with two active experts per image (rows per dataset sum to 200%).

	Dataset					
Expert	BIG	DAVIS585	ECSSD	VOC	MSRA-B	Mean
Edge-Distance	46.19%	**46.07%**	84.57%	**81.77%**	**60.38%**	**63.79%**
Depth	25.13%	22.65%	3.43%	8.92%	4.70%	12.97%
Color	6.00%	5.30%	9.77%	10.13%	9.65%	8.17%
Superpixels	4.21%	10.17%	42.10%	19.42%	37.35%	22.65%
Contrast	16.72%	23.33%	16.90%	29.79%	19.62%	21.27%
Brightness	**52.73%**	35.21%	11.40%	21.41%	13.31%	26.81%
Texture	12.68%	16.24%	7.30%	21.08%	7.85%	13.03%
Object Coverage	13.37%	18.03%	14.40%	14.41%	15.59%	15.16%
Weighted Coverage	12.68%	12.91%	2.10%	3.82%	2.81%	6.86%
SAM Box Prompt	10.30%	10.09%	8.03%	10.64%	7.35%	9.28%

R. Leave-One-Out Expert Ablation

For PromptMoE, we selected 10 experts spanning a variety of image-processing cues. While including additional experts may improve performance, particularly for domain-specific tasks requiring specialized cues (e.g., camouflaged objects), our experts were selected based on a thorough review of the image-processing literature. Of these 10 experts, some overlap exists, such as color and brightness similarity, as shown in Tab. 23. However, each expert is the strongest single cue for some masks, motivating the inclusion of each. To further evaluate expert necessity, we conducted a leave-one-out ablation study, evaluating each com-

bination of nine experts with one removed. Compared to to the 10-expert baseline (+6.24/+8.99), removing each expert reduces performance. Removing *Depth* yields the smallest drop (+6.22/+8.98), while *Brightness* yields the largest (+6.12/+8.92). Since removing any expert lowers performance, this further validates the individual contributions and utility of each expert.