Learning from Adversity: Semantic-Aware Mask Refinement through Adversarial Perturbation

Beomyoung Kim[1,2] and Sung Ju Hwang[2,3]

[1]NAVER Cloud [2]KAIST [3]DeepAuto.ai
{beomyoung.kim,sungju.hwang}@kaist.ac.kr

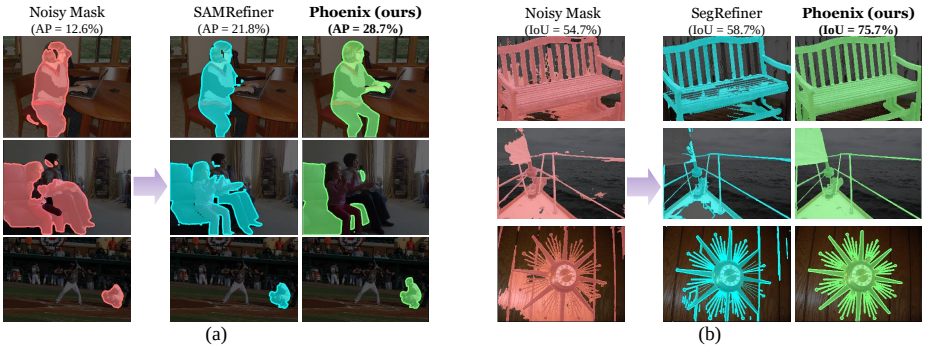
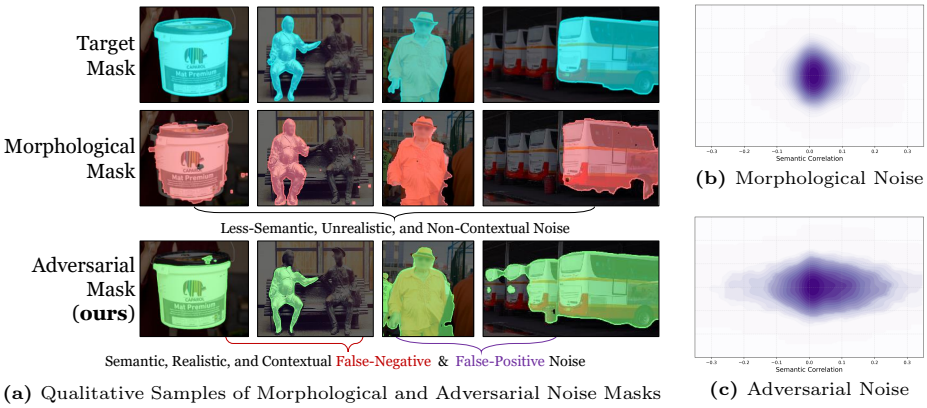


Fig. 1: Phoenix refines coarse segmentation masks into high-quality ones. *Mask refinement* corrects the boundary and structural errors of a noisy mask from an off-the-shelf model, without retraining it. On (a) instance and (b) fine-grained segmentation, we show the noisy input, recent refiners, and Phoenix; annotated values are average AP / IoU on a representative benchmark. Trained on semantic-aware adversarial noise, Phoenix best recovers complex structures, thin parts, and precise boundaries.

Abstract. Despite significant advances in image segmentation, even state-of-the-art models produce masks with imperfect boundaries, semantic inconsistencies, and structural errors. Mask refinement addresses these limitations, yet current approaches rely on simplistic synthetic noise that fails to capture the complex error patterns of real segmentation models. We introduce Phoenix, a novel framework that leverages adversarial learning to generate semantically meaningful noise patterns and contrastive learning to model refinement relationships. Our approach consists of two key innovations: (1) Adversarial Mask Perturbation, which employs embedding attacks to create semantic-aware noise that mimics real segmentation errors, and (2) Contrastive Mask Refinement Learning, which establishes a tri-directional framework that ensures feature consistency within semantic regions while maintaining separation between classes. Experiments demonstrate that Phoenix significantly outperforms existing methods across diverse tasks, while consistently enhancing state-of-the-art segmentation models with substantial improvements. Our code and project page are publicly available at https://phoenix-eccv26.github.io.



(a) Qualitative Samples of Morphological and Adversarial Noise Masks

(b) Morphological Noise

(c) Adversarial Noise

Fig. 2: Qualitative Comparison of noise patterns (a) between morphological and our adversarial noise masks. (b, c) Distribution of the Pearson correlation between noise location and image edge/texture maps on LVIS val. (b) Morphological noise is narrowly concentrated near 0 (semantically uncorrelated), while (c) our adversarial noise spans $[-0.6, 0.8]$, where positive values indicate alignment with semantic structures and negative values indicate alignment with homogeneous regions, mirroring the diverse error patterns of real models.

1 Introduction

Image segmentation provides pixel-level understanding crucial for numerous applications. Despite remarkable progress in segmentation architectures, even state-of-the-art models [7, 20, 46] exhibit persistent limitations: imprecise boundaries at complex contours, semantic confusion between similar objects, and structural inconsistencies that violate object integrity. These errors stem from fundamental challenges that remain despite extensive model scaling and data collection.

This circumstance establishes mask refinement as a distinct and versatile complementary approach to conventional segmentation methods, directly addressing their systematic limitations. It enhances model performance without architectural changes, making it valuable in cases where model updates are infeasible, such as due to proprietary restrictions, computational limitations, or data collection costs. Furthermore, mask refinement is pivotal in label-efficient learning, including semi-supervised [16, 45] and weakly-supervised settings [18, 40], transforming low-quality pseudo-labels into reliable supervision, which maximizes learning outcomes even with limited annotations.

The construction of realistic noisy and clean mask pairs is central to effective refinement learning. Recent efforts [28, 38] have advanced mask refinement through various approaches, yet significant limitations persist. Methods like SegFix [48] and SegRefiner [42] rely on synthetic noise generated through morphological operations, producing simplistic, spatially random perturbations that often fail to capture the structured, context-dependent errors of real neural networks. This fundamental limitation restricts their ability to address the complex challenges in real-world segmentation tasks, as shown in Figure 1.

"***What doesn't kill you makes you stronger.***" This wisdom reflects how adversity can become a catalyst for growth, a principle we translate from philosophy to algorithm design. In natural systems, adaptation occurs in response to meaningful challenges. Similarly, the effective learning system emerges from facing realistic, meaningful obstacles rather than artificial ones. Drawing inspiration from this, we introduce a novel framework for semantic-aware mask refinement through adversarial perturbation. We dub our framework ***Phoenix*** since it transforms challenging noise patterns into superior refinement capabilities, mirroring how the mythical bird rises stronger from the ashes.

The Phoenix framework builds upon two key innovations. First, Adversarial Mask Perturbation (AMP) employs adversarial embedding attacks to generate semantically meaningful, contextually aware noise patterns. By optimizing against the model's learned representations, AMP creates perturbations that concentrate precisely where real segmentation models struggle most, such as at semantically challenging boundaries and ambiguous regions, as shown in Figure 2. Notably, this approach offers controllability over noise generation patterns while maintaining computational efficiency. Second, our Contrastive Mask Refinement Learning (CMRL) explicitly models the relationships between ground truth, noisy input, and current prediction. Unlike conventional pixel-wise objective functions, this tri-directional contrastive mechanism promotes clear separation between foreground and background features while ensuring consistency within semantic regions, which is tailored to the mask refinement task.

Our experiments demonstrate that Phoenix significantly outperforms existing refinement methods [28, 38, 42, 48] across diverse tasks. When refining masks from semi-supervised and weakly-supervised models, Phoenix achieves absolute gains of up to +16.1% in AP^{mask}. Applied to state-of-the-art segmentation models, it consistently improves performance across diverse architectures. Furthermore, we demonstrate Phoenix's effectiveness in fine-grained mask refinement tasks and its potential for self-supervised refinement without ground-truth annotations.

2 Related Work

2.1 Image Segmentation

Image segmentation has evolved significantly with the advent of deep learning. Early approaches like FCN [29] and U-Net [36] established fully-convolutional architectures that became foundational for subsequent research. Performance improvements followed through innovations in feature extraction (*e.g.*, DeepLab [4], PSPNet [49]) and, more recently, with transformer-based architectures (*e.g.*, SegFormer [46], Mask2Former [7]) that leverage global context modeling. The recent Segment Anything Model (SAM) [20] represents a significant advance in general-purpose segmentation through its prompt-based inference and zero-shot capabilities.

Despite these advances, generating high-quality segmentation masks remains challenging, particularly at object boundaries and in complex scenes. This chal-

lenge is magnified in the semi-supervised [45] and weakly semi-supervised [16] settings, where initial segmentation predictions are often coarse and contain substantial noise.

2.2 Mask Refinement

Mask refinement addresses the limitations of primary segmentation models by enhancing mask quality through post-processing. Early approaches employed traditional techniques like conditional random fields (CRF) [22] and graph cuts [2] to improve boundary adherence. These methods often rely on handcrafted features and struggle with semantically complex scenes. Learning-based refinement has gained traction with various approaches. SegFix [48] employs residual correction for boundaries, BPR [38] introduces a boundary patch refinement that focuses on local regions around object boundaries. SegRefiner [42] adopts a diffusion process for mask refinement modeling with a two-stage approach combining coarse prediction and fine-grained refinement. Recently, SAMRefiner [28] leverages SAM's zero-shot capabilities through visual prompt engineering for mask refinement. Despite its effectiveness with a training-free setup, it relies solely on fixed pre-trained representations that weren't optimized for the refinement task. In contrast, Phoenix utilizes SAM's architecture with efficient fine-tuning techniques specifically designed to address key refinement challenges, such as correction pattern learning and boundary precision.

The primary challenge in effective mask refinement modeling lies in generating representative training data that captures realistic error patterns. Existing approaches [42, 48] predominantly rely on morphological perturbations of ground-truth masks to simulate noise, such as random boundary perturbations with dilation and erosion operations and region modifications. These methods produce synthetic noise patterns that fail to capture the semantic nature of errors in real segmentation models, which are highly structured and context-dependent.

2.3 Adversarial Learning

Adversarial learning has primarily focused on attack and defense mechanisms for model robustness [9,31]. Adversarial attacks on segmentation models [1] and black-box perturbation methods [13] demonstrate how adversarial techniques can expose model vulnerabilities through carefully crafted perturbations. However, existing work treats adversarial perturbations as destructive tools for testing rather than constructive mechanisms for training data generation. In contrast, we repurpose adversarial perturbation from a destructive testing tool into a constructive data-generation mechanism for mask refinement learning, operating in the embedding space to generate semantically meaningful noise patterns that mimic realistic segmentation errors instead of merely maximizing prediction failures.

2.4 Contrastive Learning

Contrastive learning has achieved remarkable success in representation learning [5, 11] with recent extensions to dense prediction tasks. [43] demonstrate that contrastive objectives in feature space can improve dense prediction quality through alignment and uniformity principles. Pixel-wise contrastive methods [44, 47] have been developed for semantic segmentation pre-training and unsupervised representation learning. However, existing approaches primarily focus on learning general visual representations or establishing class boundaries. In contrast, our Contrastive Mask Refinement Learning introduces a novel tri-directional framework specifically designed for modeling the refinement relationship between noisy inputs, predictions, and ground truth masks, explicitly capturing the transformation from incorrect to correct predictions through self-improvement regularization.

3 Methodology

3.1 Problem Formulation

Mask refinement aims to transform a noisy or coarse segmentation mask into a high-quality mask that accurately delineates object boundaries and semantic regions. Formally, given an input image $\mathcal{I} \in \mathbb{R}^{H\times W\times 3}$ and a corresponding noisy mask $\mathcal{M}_n \in \{0,1\}^{H\times W}$, the mask refiner f generates a refined mask $\mathcal{M}_r \in \{0,1\}^{H\times W}$ such that: $\mathcal{M}_r = f(\mathcal{I}, \mathcal{M}_n; \theta)$, where θ represents the parameters of the refiner model, and H and W denote the height and width of the image.

3.2 Framework Overview

Figure 3 illustrates our Phoenix framework consisting of two key innovations: (1) Adversarial Mask Perturbation (AMP) and (2) Contrastive Mask Refinement Learning (CMRL). Phoenix builds upon pre-trained SAM [20] with fine-tuning only the lightweight decoder. The encoder f_{enc} takes an input image $\mathcal{I}$ and extracts image embeddings $\mathbf{E}_{img}$. The decoder f_{dec} then processes these embeddings along with the noisy mask $\mathcal{M}_n$ to produce the refined mask $\mathcal{M}_r$. In addition, we extract point and box prompts from the noisy mask and incorporate them into the decoder's input as visual prompt embeddings $\mathbf{E}_v$, following the prompt sampling strategy used in [28].

3.3 Adversarial Mask Perturbation (AMP)

Limitations of Morphological Noise Approaches. Existing mask refinement methods [42, 48] predominantly rely on morphological perturbations (erosion, dilation, boundary modifications) to create synthetic noise from ground-truth masks. These approaches have several limitations, as evident in Figure 2a: (1) They produce unrealistic noise patterns that fail to capture the semantic

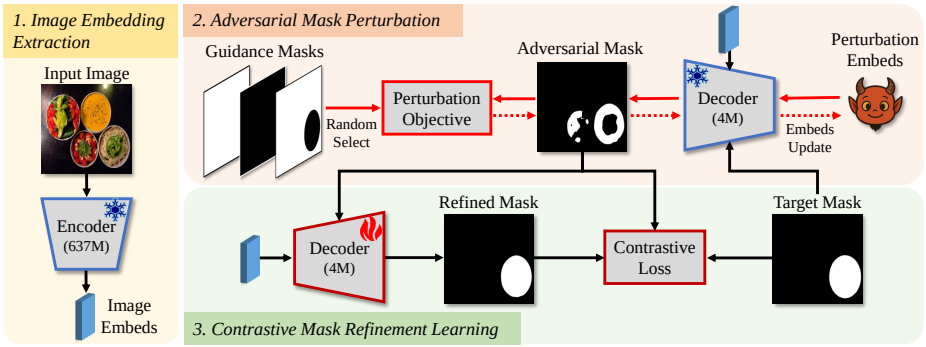


Fig. 3: Overview of our Phoenix framework. The pipeline consists of three main components: (1) Image Embedding Extraction using SAM's encoder, (2) Adversarial Mask Perturbation that generates realistic noise patterns through adversarial embedding attacks, and (3) Contrastive Mask Refinement Learning that uses the tri-directional relationships between masks to improve refinement quality.

nature of errors in real segmentation models, as morphological operations create structurally simplistic and spatially random noise, (2) They generate noise with limited diversity, unable to represent the wide range of failure modes in modern segmentation models, and (3) They are contextually blind, with perturbations operating independently of image content, making them unable to simulate errors from semantic confusion between similar objects.

Proposed Adversarial Approach. We introduce Adversarial Mask Perturbation (AMP), which leverages adversarial embedding attacks to generate semantically meaningful, contextually aware noise patterns. Given a target (ground-truth) mask $\mathcal{M}_t$, we inject learnable perturbation embeddings $\mathbf{E}_p \in \mathbb{R}^{P\times C}$ into the decoder f_{dec} alongside visual prompt embeddings $\mathbf{E}_v$ derived from $\mathcal{M}_t$, where P is the number of embeddings and C is the embedding dimension. Importantly, the perturbation embeddings $\mathbf{E}_p$ are used exclusively for generating noisy masks during data augmentation without affecting the pretrained decoder parameters. The decoder f_{dec} remains frozen during perturbation, and the $\mathbf{E}_p$ are not involved in actual training or inference of the decoder.

Unlike conventional adversarial attacks [1,9,31] that aim to maximize classification error by perturbing input images, our approach repurposes adversarial techniques as constructive tools for noise generation. Specifically, we adapt the FGSM [9] to operate in the embedding space rather than image pixel space: $\mathbf{E}_p \leftarrow \mathbf{E}_p + \alpha \cdot \text{sign}(\nabla_{\mathbf{E}_p}\mathcal{L}_{adv})$, where α controls the perturbation magnitude and $\mathcal{L}_{adv}$ is an adversarial objective. By operating in the embedding space rather than input image space, we achieve both higher computational efficiency and high-level semantic perturbation [13].

Controllable Noise Generation. The *Guidance Mask* ($\mathcal{M}_g$) determines the semantic direction of perturbation by modifying the adversarial objective:$\mathcal{L}_{adv} = -\mathcal{D}(f_{dec}(\mathbf{E}_{img}, [\mathbf{E}_p; \mathbf{E}_v]), \mathcal{M}_g)$, where $\mathcal{D}$ represents an adversarial criterion (*e.g.*,

Algorithm 1 Adversarial Mask Perturbation

Require: Image embeds $\mathbf{E}_{img}$, perturbation embeds $\mathbf{E}_p$, visual prompt embeds $\mathbf{E}_v$, target mask $\mathcal{M}_t$, guidance mask $\mathcal{M}_g$, adversarial criterion $\mathcal{D}$, initial step size α_0, IoU threshold τ, margin ϵ, maximum inner iterations N.
Ensure: Noisy mask $\mathcal{M}_n$ with IoU $[\tau, \tau + \epsilon]$

$\alpha = \alpha_0$
repeat
  for $i = 1$ to N **do**
    $\mathcal{M}_n = f_{dec}(\mathbf{E}_{img}, [\mathbf{E}_p; \mathbf{E}_v])$
    $iou = \text{compute\_IoU}(\mathcal{M}_n, \mathcal{M}_t)$
    if $iou < \tau + \epsilon$ **then**
      break
    end if
    Save current state: $\mathbf{E}_p^{prev} = \mathbf{E}_p$
    $\mathcal{L}_{adv} = -\mathcal{D}(\mathcal{M}_n, \mathcal{M}_g)$
    Compute gradient $\nabla_{\mathbf{E}_p} \mathcal{L}_{adv}$
    $\mathbf{E}_p = \mathbf{E}_p + \alpha \cdot \text{sign}(\nabla_{\mathbf{E}_p} \mathcal{L}_{adv})$
  end for
  if $iou \geq \tau$ **then**
    return $\mathcal{M}_n$ {Target IoU achieved}
  end if
  $\alpha = \alpha/10$ {Decay step size}
  Restore previous state: $\mathbf{E}_p = \mathbf{E}_p^{prev}$
until maximum decay steps reached
return $\mathcal{M}_n$

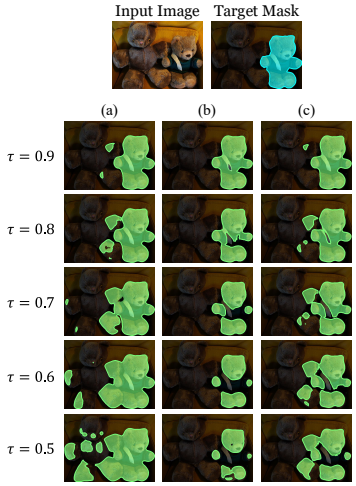


Fig. 4: Qualitative Samples of generated noisy masks according to the IoU threshold τ and guidance mask (a) expansion guide, (b) contraction guide, and (c) inversion guide.

Dice or MSE loss). Three primary configurations yield distinct noise patterns, as shown in Figure 4: (1) *Expansion Guide* ($\mathcal{M}_g = \mathbf{1}$, all ones) generates false-positive errors by pushing boundaries outward, (2) *Contraction Guide* ($\mathcal{M}_g = \mathbf{0}$, all zeros) creates false-negative errors that mimic under-segmentation behaviors, (3) *Inversion Guide* ($\mathcal{M}_g = 1 - \mathcal{M}_t$) produces a balanced distribution of both error types.

Furthermore, to automatically calibrate the noise magnitude, Algorithm 1 implements an adaptive threshold-guided noise generation approach that adjusts the perturbation strength based on the IoU threshold parameter τ. Given initial step size α_0, IoU threshold τ, margin ϵ, and maximum inner iterations N, it iteratively updates $\mathbf{E}_p$ according to the FGSM rule, the process continues until the IoU between the generated noisy mask and the target mask falls within the range $[\tau, \tau + \epsilon]$. If this condition is not met until N iterations, it reduces the step size $\alpha \leftarrow \alpha/10$ and restarts the process. As shown in Figure 4, the IoU threshold τ enables control over noise intensity: lower thresholds generate more aggressive noise patterns, while higher thresholds produce subtle noise.

Semantic Distribution: Principled Foundation and Design Choices. For a pretrained decoder f_{dec} with fixed parameters θ_{dec}, the embedding gradient magnitude varies across spatial locations based on the model's uncertainty. Formally, this gradient magnitude relates to the local classification uncertainty:

$$\left\|\nabla_{\mathbf{E}_p} f_{\text{dec}}(\mathbf{E}_{\text{img}}, [\mathbf{E}_p; \mathbf{E}_v]; \theta_{\text{dec}})\right\|_2 \propto -\log p(y \mid \mathbf{E}_{\text{img}}, [\mathbf{E}_p; \mathbf{E}_v]; \theta_{\text{dec}}) \quad (1)$$

, where $p(y|\mathbf{E}_{img}, [\mathbf{E}_p; \mathbf{E}_v]; \theta_{dec})$ represents the decoder's confidence in its prediction y at each spatial location [15]. Consequently, our FGSM-based update intrinsically amplifies perturbations in regions of high uncertainty, precisely where segmentation models struggle, producing semantically meaningful noise patterns.

We make the scope of this analysis explicit, separating principled foundations from empirically-validated design choices. The perturbation *direction* is principled: it follows the FGSM formulation and inherits the gradient–uncertainty relationship from Bayesian deep learning [15], where the gradient magnitude grows with prediction uncertainty (empirically verified in the embedding space in the supplementary material). In contrast, the perturbation *magnitude* (controlled by τ) and the *semantic direction* (selected by the guidance mask $\mathcal{M}_g$) are design choices that we validate empirically (Tables 4a and 4b), rather than closed-form guarantees. Likewise, our claim that AMP better matches realistic segmentation errors is supported *empirically* (Tables 4d and 4e), not as a formal proof. Throughout, "semantic" refers to the *model's* learned semantics, i.e., where the model is most uncertain, rather than human-perceptual semantics.

We quantify the semantic nature of our generated noise through semantic correlation analysis between the perturbation magnitude and image feature gradients using the LVIS dataset [10]. This analysis computes Pearson correlation [33] between noise spatial distribution and semantic features extracted from edge detection and texture maps to measure how perturbations relate to image semantics (detailed in the supplementary material). As shown in Figure 2c, adversarial noise exhibits a broad distribution of semantic correlations spanning $[-0.6, 0.8]$, indicating its ability to capture both semantically important regions (positive values) and homogeneous areas (negative values). In contrast, morphological noise (Figure 2b) shows a narrow distribution concentrated around zero, confirming its semantic-agnostic nature. This quantitative analysis validates that AMP produces noise patterns that align with the complex error distributions.

Computational Efficiency. Our implementation maintains high computational efficiency by reusing image embeddings from the encoder's single forward pass (637M parameters for ViT-H). Since AMP operates only on the lightweight decoder (4M parameters, 1.01 GFlops), each perturbation update requires only 6 ms on a V100 GPU, enabling efficient noise generation.

3.4 Contrastive Mask Refinement Learning (CMRL)

Motivation. Traditional refinement approaches [38, 42, 48] primarily rely on pixel-wise classification, which treats each pixel independently. The fundamental challenge in mask refinement lies in modeling complex error patterns and their

relationship to correct image contexts. Building on advances in contrastive learning, we propose a Contrastive Mask Refinement Learning (CMRL) approach that explicitly models the relationships between ground truth, noisy input, and model output masks. While conventional contrastive methods [5,11] focus on representation learning by contrasting positive and negative pairs, our tri-directional approach establishes a more complex and unique relationship structure across three different masks and is designed to explicitly guide the mask refinement learning, which is tailored specifically for the mask refinement task.

Formulation. CMRL is designed with a tri-directional contrastive framework to address three objectives: (1) maintaining clear separation between foreground and background features, (2) ensuring consistency among features within the same semantic region, and (3) enabling self-improvement through bootstrapping from successful refinements. First, CMRL categorizes pixels into six distinct regions based on their classification in target ($\mathcal{M}_t$), noisy ($\mathcal{M}_n$), and refined ($\mathcal{M}_r$) masks:

$$\begin{aligned}
\mathcal{T}_{fg} &= (\mathcal{M}_t = 1) \wedge (\mathcal{M}_n = 1) \wedge (\mathcal{M}_r = 1), \quad \mathcal{T}_{bg} = (\mathcal{M}_t = 0) \wedge (\mathcal{M}_n = 0) \wedge (\mathcal{M}_r = 0) \\
\mathcal{S}_{fg} &= (\mathcal{M}_t = 1) \wedge (\mathcal{M}_n = 0) \wedge (\mathcal{M}_r = 1), \quad \mathcal{S}_{bg} = (\mathcal{M}_t = 0) \wedge (\mathcal{M}_n = 1) \wedge (\mathcal{M}_r = 0) \\
\mathcal{F}_{fg} &= (\mathcal{M}_t = 1) \wedge (\mathcal{M}_n = 0) \wedge (\mathcal{M}_r = 0), \quad \mathcal{F}_{bg} = (\mathcal{M}_t = 0) \wedge (\mathcal{M}_n = 1) \wedge (\mathcal{M}_r = 1)
\end{aligned}$$

where $\mathcal{T}$, $\mathcal{S}$, and $\mathcal{F}$ denote true, success, and failure regions, with fg and bg indicating foreground and background. This categorization creates a natural curriculum where the model progressively refines its predictions by learning from both initially correct regions and its own successful refinements.

Notation. Let $\mathbf{F}$ denote upsampled image embeddings. We apply projector g consisting of a 3-layer MLP to obtain projection feature maps $\mathbf{p} = g(\mathbf{F}) \in \mathbb{R}^{c \times h \times w}$. For each position $i \in \Omega = \{1, \ldots, h \times w\}$, $\mathbf{p}_i \in \mathbb{R}^c$ denotes the feature vector at position i. The expectation $\mathbb{E}_{i \in \mathcal{R}}[\cdot]$ denotes uniform sampling over pixels in region $\mathcal{R}$. The similarity function $\text{sim}(\mathbf{p}_i, \mathbf{p}_j) = \mathbf{p}_i^\top \mathbf{p}_j$ computes cosine similarity between L2-normalized features. Unlike conventional segmentation losses that operate in the pixel space, our contrastive framework operates in the feature space, enabling the model to learn rich feature representations that capture contextual information [43].

Intra-Class Feature Consistency aims to create coherent feature representations within each semantic class, ensuring that all parts of the same object share similar feature characteristics regardless of their visual appearance. For foreground features, the intra-class loss is defined as:

$$\mathcal{L}_{intra}^{fg} = -\mathbb{E}_{i \in \mathcal{F}_{fg}} \left[\log \frac{\sum_{j \in \mathcal{S}_{fg} \cup \mathcal{T}_{fg}} \exp(\text{sim}(\mathbf{p}_i, \mathbf{p}_j)/\tau)}{\sum_{k \in \Omega}} \exp(\text{sim}(\mathbf{p}_i, \mathbf{p}_k)/\tau) \right] \tag{2}$$

where τ is a temperature parameter. Following the InfoNCE contrastive loss [32], this loss maximizes the similarity between foreground failure features and correct foreground features while implicitly pushing away features from other regions. A similar loss $\mathcal{L}_{intra}^{bg}$ is computed for background features, and the total intra-class loss is $\mathcal{L}_{intra} = \mathcal{L}_{intra}^{fg} + \mathcal{L}_{intra}^{bg}$.

Inter-Class Feature Contrast enforces clear separation between foreground and background features, particularly in error-prone regions, by maximizing the distance between the two features:

$$\mathcal{L}_{inter}^{fg \to bg} = \mathbb{E}_{i \in \mathcal{F}_{fg}} \left[\log \sum_{j \in \mathcal{F}_{bg} \cup \mathcal{S}_{bg} \cup \mathcal{T}_{bg}} \exp(\text{sim}(\mathbf{p}_i, \mathbf{p}_j)/\tau) \right] \tag{3}$$

It pushes foreground failure features away from all background features. Likewise, $\mathcal{L}_{inter}^{bg \to fg}$ is computed. By maximizing the feature distance between different classes, this bidirectional repulsion reshapes the feature space to create clearer decision boundaries between foreground and background. The total inter-class loss is $\mathcal{L}_{inter} = \mathcal{L}_{inter}^{fg \to bg} + \mathcal{L}_{inter}^{bg \to fg}$.

Self-Improvement Regularization leverages successfully refined regions to guide the improvement of currently unrefined errors, creating a learning pathway from failure to success within the same image. This component enables the model to learn from its own successful corrections, addressing the unique challenge of transforming incorrect predictions into correct ones.

$$\mathcal{L}_{self} = -\mathbb{E}_{i \in \mathcal{F}_{fg} \cup \mathcal{F}_{bg}} \left[\log \frac{\sum_{j \in \mathcal{S}_{fg} \cup \mathcal{S}_{bg}}}{\exp} (\text{sim}(\mathbf{p}_i, \mathbf{p}_j)/\tau) \sum_{k \in \Omega} \exp(\text{sim}(\mathbf{p}_i, \mathbf{p}_k)/\tau) \right] \tag{4}$$

This loss encourages features from current failure regions ($\mathcal{F}_{fg} \cup \mathcal{F}_{bg}$) to resemble those from successfully refined regions ($\mathcal{S}_{fg} \cup \mathcal{S}_{bg}$) within the same image. Unlike the other components that focus on static correctness, this loss explicitly models the transformation from incorrect to correct predictions, creating a bootstrapping mechanism where the model learns from its own successful refinements to improve regions that are still incorrectly classified.

Loss Function. Our final CMRL loss combines all three objectives with weighting parameters:

$$\mathcal{L}_{CMRL} = \lambda_{intra} \cdot \mathcal{L}_{intra} + \lambda_{inter} \cdot \mathcal{L}_{inter} + \lambda_{self} \cdot \mathcal{L}_{self} \tag{5}$$

where λ_{intra}, λ_{inter}, and λ_{self} balance the respective objectives. This contrastive loss is combined with a traditional segmentation loss to produce the final training objective.

4 Experiments

4.1 Experimental Setup

Datasets. For general object mask refinement, we train Phoenix on the LVIS [10] dataset following the same setup as SegRefiner [42]. We evaluate on masks from both low-quality and high-quality segmentation models, refining coarse masks generated by semi-supervised (NB [45]) and weakly semi-supervised (PointWSSIS [16]) methods by measuring the quality of pseudo labels on the COCO

Table 1: Performance of refined masks on COCO train5K using LVIS annotations. We denote full mask annotations as $\mathcal{F}$, unlabeled data as $\mathcal{U}$, and (object center) point annotations as $\mathcal{P}$.

(a) Semi-Supervised, NB [45]

Methods	Annotations	AP^{mask}	$AP^{boundary}$
NB	$\mathcal{F}$ 1% + $\mathcal{U}$ 99%	5.1	1.9
+SegRefiner		6.2	3.8
+SAMRefiner		8.1	6.0
+Phoenix (ours)		**9.8** (+4.7)	**8.1** (+6.2)
NB	$\mathcal{F}$ 5% + $\mathcal{U}$ 95%	18.5	9.7
+SegRefiner		20.4	14.0
+SAMRefiner		23.8	18.5
+Phoenix (ours)		**26.7** (+8.2)	**22.2** (+12.5)
NB	$\mathcal{F}$ 10% + $\mathcal{U}$ 90%	22.6	12.7
+SegRefine		25.0	17.9
+SAMRefiner		28.2	22.1
+Phoenix (ours)		**30.8** (+8.2)	**25.7** (+13.0)

(b) Weakly Semi-Supervised, PointWSSIS [16]

Methods	Annotations	AP^{mask}	$AP^{boundary}$
PointWSSIS	$\mathcal{F}$ 1% + $\mathcal{P}$ 99%	12.6	6.3
+SegRefiner		14.7	9.5
+SAMRefiner		21.8	16.4
+Phoenix (ours)		**28.7** (+16.1)	**23.6** (+17.3)
PointWSSIS	$\mathcal{F}$ 5% + $\mathcal{P}$ 95%	25.7	16.4
+SegRefiner		27.6	20.2
+SAMRefiner		32.8	26.2
+Phoenix (ours)		**36.3** (+10.6)	**30.2** (+13.8)
PointWSSIS	$\mathcal{F}$ 10% + $\mathcal{P}$ 90%	30.2	20.4
+SegRefiner		31.7	23.8
+SAMRefiner		36.6	29.7
+Phoenix (ours)		**38.9** (+8.7)	**32.6** (+12.2)

Table 2: Performance of refined masks on COCO validation set using LVIS annotations.

(a) Results on Mask R-CNN

Method	AP^{mask}	$AP^{boundary}$
MRCNN(RN50)	39.8	27.3
+SegFix	40.6	29.1
+BPR	41.0	30.4
+SegRefiner	41.9	32.6
+SAMRefiner	45.3	35.9
+Phoenix (ours)	**46.9** (+7.1)	**38.8** (+11.5)
MRCNN(RN101)	41.6	29.0
+SegFix	42.2	30.6
+BPR	42.8	32.0
+SegRefiner	43.6	34.1
+SAMRefiner	46.6	36.9
+Phoenix (ours)	**48.1** (+6.5)	**39.8** (+10.8)

(b) Results on State-of-the-art Segmentation Models

Method	AP^{mask}	$AP^{boundary}$	Method	AP^{mask}	$AP^{boundary}$
SOLO	37.4	24.7	CondInst	39.8	29.2
+SegRefiner	40.5	31.3	+SegRefiner	41.1	32.2
+SAMRefiner	44.1	34.2	+SAMRefiner	45.2	35.8
+Phoenix (ours)	**46.2** (+8.8)	**37.7** (+13.0)	+Phoenix (ours)	**46.7** (+6.9)	**38.6** (+9.4)
RefineMask	41.2	30.5	Mask2Former	46.8	37.0
+SegRefiner	41.9	33.0	+SegRefiner	47.4	38.8
+SAMRefiner	44.7	35.3	+SAMRefiner	49.0	39.0
+Phoenix (ours)	**46.0** (+4.8)	**37.9** (+7.4)	+Phoenix (ours)	**50.6** (+3.8)	**42.1** (+5.1)
ViTDet	54.6	42.5	MaskDINO	56.8	46.5
+SegRefiner	55.5	46.0	+SegRefiner	57.0	47.7
+SAMRefiner	55.8	46.2	+SAMRefiner	57.0	47.4
+Phoenix (ours)	**56.3** (+1.7)	**46.7** (+4.2)	+Phoenix (ours)	**58.0** (+1.2)	**48.6** (+2.1)

Train5K dataset [16]. We also evaluate on masks produced by various instance segmentation models on the COCO [27] validation set. Separately, for the fine-grained segmentation task, we train Phoenix on DIS5K [35] and ThinObject-5K [25] datasets and evaluate it on the DIS task [35], which demands precise boundary delineation for thin structures and complex topologies.

Evaluation Metrics. We use Average Precision (AP) and boundary AP ($AP^{boundary}$) [6] to evaluate instance segmentation quality, and Intersection over Union (IoU) and Boundary $\mathcal{F}$ measure [34] for fine-grained segmentation tasks.

Implementation Details. Our mask refiner builds upon the pre-trained SAM [20] with the ViT-H [8] backbone. During training, we freeze the encoder (637M parameters) and fine-tune only the decoder (4M parameters). We optimize the model using AdamW [30] with an initial learning rate of 1×10^{-4} and cosine decay scheduling. Training proceeds for 10K iterations with a total batch size of 16 on 8 V100 GPUs, requiring less than 10 hours of total training time. For the adversarial mask perturbation process, we use Dice loss as the perturbation objective with the maximum inner iterations N of 10, the initial step size

Table 3: Performance of refined masks on the DIS task using coarse masks from 4 different models. SAMRefiner† means using the HQ-SAM [14] backbone for finer mask prediction.

Methods	DIS-VD		DIS-TE1		DIS-TE2		DIS-TE3		DIS-TE4		*Average*	
	IoU	Boundary $\mathcal{F}$	IoU	Boundary $\mathcal{F}$	IoU	Boundary $\mathcal{F}$	IoU	Boundary $\mathcal{F}$	IoU	Boundary $\mathcal{F}$	IoU	Boundary $\mathcal{F}$
U-Net	54.8	67.0	44.1	59.2	54.4	66.1	59.3	72.0	61.1	77.8	54.7	68.4
+SAMRefiner†	63.7	72.4	56.5	74.3	67.1	76.3	69.3	74.4	64.8	64.8	64.3	72.4
+SegRefiner	58.7	71.0	47.0	63.4	58.1	70.7	63.4	75.9	66.3	81.8	58.7	72.6
+Phoenix (ours)	**74.9** (+20.1)	**84.2** (+17.2)	**69.6** (+25.5)	**83.3** (+24.1)	**79.3** (+24.9)	**85.8** (+19.7)	**79.4** (+20.1)	**85.8** (+13.8)	**75.2** (+14.1)	**82.9** (+5.1)	**75.7** (+21.0)	**84.4** (+16.0)
PSPNet	56.4	69.3	47.6	66.8	57.7	70.5	59.9	71.8	57.6	70.4	55.8	69.8
+SAMRefiner†	64.2	71.7	58.8	76.4	67.2	76.6	68.0	72.5	63.2	62.9	64.3	72.0
+SegRefiner	61.9	73.7	50.4	69.0	62.0	74.3	65.3	77.5	66.7	80.3	61.3	75.0
+Phoenix (ours)	**75.7** (+19.3)	**84.8** (+15.5)	**70.3** (+22.7)	**84.0** (+17.2)	**79.5** (+21.8)	**86.0** (+15.5)	**79.8** (+19.9)	**86.2** (+14.4)	**74.4** (+16.8)	**82.8** (+12.4)	**75.9** (+20.1)	**84.8** (+15.0)
HRNet	61.0	74.2	51.0	67.1	61.2	73.6	64.4	77.9	64.7	82.0	60.5	74.9
+SAMRefiner†	67.5	74.2	59.7	76.6	68.1	77.9	72.6	76.0	65.8	65.0	66.8	73.9
+SegRefiner	64.4	76.1	52.9	68.7	64.7	75.7	67.6	79.8	70.5	84.6	64.0	77.0
+Phoenix (ours)	**76.5** (+15.5)	**85.8** (+11.6)	**69.6** (+18.6)	**83.4** (+16.3)	**78.1** (+16.9)	**86.4** (+12.8)	**80.7** (+16.3)	**86.6** (+8.7)	**74.9** (+10.2)	**83.4** (+1.4)	**76.0** (+15.5)	**85.1** (+10.2)
ISNet	67.1	79.8	56.2	74.9	67.1	78.4	69.9	81.2	70.1	83.8	66.1	79.6
+SAMRefiner†	68.1	75.2	60.6	78.4	69.4	78.2	71.7	74.8	66.0	64.0	67.1	74.1
+SegRefiner	68.3	80.2	56.9	75.0	67.8	79.0	70.9	81.9	72.2	85.2	67.2	80.3
+Phoenix (ours)	**76.7** (+9.6)	**86.1** (+6.3)	**71.4** (+17.7)	**85.3** (+10.4)	**79.3** (+12.2)	**86.7** (+8.3)	**80.2** (+10.3)	**87.0** (+5.8)	**75.3** (+5.2)	**84.2** (+0.4)	**76.6** (+11.0)	**85.9** (+6.3)

α_0 of 0.01, and the margin ϵ of 5%. The IoU threshold τ is randomly sampled from a uniform distribution between 0.3 and 0.9, and the guidance mask $\mathcal{M}_g$ is randomly chosen among expansion, contraction, and inversion guides. For contrastive loss weightings, we set $\lambda_{\text{intra}} = 0.4$, $\lambda_{\text{inter}} = 0.4$, and $\lambda_{\text{self}} = 0.2$ by default. The contrastive loss $\mathcal{L}_{CMRL}$ is added to the loss function used in SAM, which is the combination of Dice and Focal [26] losses.

4.2 Results on Instance Segmentation

Table 1 presents the performance of various refinement methods on coarse masks generated by semi-supervised [45] and weakly semi-supervised [16] approaches. Our method consistently outperforms previous state-of-the-art refiners across all settings, largely due to our semantic-aware noise modeling and tri-directional contrastive learning. Phoenix achieves significant improvements when refining extremely noisy masks produced with minimal supervision (1% fully-annotated and 99% point labels), with absolute gains of up to +16.1% in AP^{mask} and +17.3% in $\text{AP}^{\text{boundary}}$. Moreover, Table 2 shows our effectiveness in refining masks from various modern instance segmentation models [7, 12, 21, 23, 24, 39]. Phoenix achieves superior performance to existing refiners in all settings, demonstrating the versatility of our refinement approach across model architectures.

4.3 Results on Fine-Grained Segmentation

Table 3 presents the performance of Phoenix when applied to the challenging DIS [35] task across multiple test datasets and backbone models [36, 37, 41, 49]. The results demonstrate that our method consistently outperforms both SAMRefiner and SegRefiner by substantial margins, with average improvements ranging from 11% to 21% in IoU, highlighting of the effectiveness of our realistic and contextual noise perturbation modeling in the mask refinement task. Figure 1 provides visual comparisons between our method and existing refiners. Phoenix produces masks with significantly improved boundary adherence and structural integrity compared to both noisy inputs and competing refiners, successfully recovering challenging scenarios, such as thin structures and intricate boundaries.

4.4 Ablation Studies

We conduct extensive ablation studies to analyze the contribution of individual components and hyperparameters. For these experiments, we use low-quality masks from PointWSSIS with 1% supervision (denoted as AP^1) and high-quality masks from Mask R-CNN with ResNet-50 [12] backbone (denoted as AP^2). Additionally, we evaluate fine-grained segmentation performance on DIS-UNet (denoted as IoU^1) and DIS-ISNet (denoted as IoU^2), averaged across all the DIS test splits. **Additional ablation studies and in-depth analyses can be found in the supplementary material.**

Effect of Adversarial Mask Perturbation Parameters. Table 4a shows the impact of the IoU threshold τ on refinement performance. Higher τ values lead to better performance on high-quality masks (higher AP^2) but lower performance on low-quality masks (lower AP^1). For robustness against various noise patterns, we randomly sample τ from a uniform distribution $\mathcal{U}(0.3, 0.9)$ for each noise mask generation step during training, achieving the best balanced performance. Moreover, Table 4b presents the performance with different adversarial criteria, demonstrating our method is relatively robust to the choice of objective, with Dice loss providing the best overall balance.

Effect of Contrastive Loss Components. Table 4c illustrates the performance impact of different contrastive loss components. We observe that both intra-class consistency ($\mathcal{L}_{intra}$) and inter-class contrast ($\mathcal{L}_{inter}$) contribute significantly to performance, with their combination yielding substantial improvements. Including all three components achieves the optimal performance.

Effect of Mask Perturbation Method. Table 4d compares different mask perturbation approaches across models. When applying adversarial perturbation to SegRefiner, its performance improves substantially. Conversely, when our Phoenix is trained with simple morphological perturbations, performance drops significantly compared to our full approach. These results highlight the critical importance of sophisticated noise generation in mask refinement learning.

Comparison with Real Model Errors. Table 4e compares our adversarially generated noise patterns with noise obtained directly from real segmentation models. While using output masks from the pre-trained model (UNet [36] or ISNet [37]) as noisy masks for training provides adequate performance, our adversarial approach significantly outperforms them. This superiority stems from two key advantages: (1) a wider diversity of error patterns, and (2) controllable perturbation magnitude.

Since there is no single ground-truth definition of a "real" segmentation error (different models fail in different ways), we assess realism through *functional substitutability*: noise that better substitutes for real errors should yield better downstream refinement. Several converging results support this view. First, the same AMP noise is causally responsible for the gains regardless of architecture: replacing SegRefiner's morphological noise with AMP improves it substantially, whereas replacing AMP with morphological noise in Phoenix degrades it sharply

Table 4: Ablation Study using instance segmentation results on 1% PointWSSIS (AP^1) and MRCNN with a ResNet-50 backbone (AP^2) and fine-grained segmentation results on DIS-UNet (IoU^1) and DIS-ISNet (IoU^2), averaged across all DIS tasks. We denote morphological perturbation as *morp* and adversarial perturbation as *adv*.

(a) τ (IoU Threshold)

τ	AP^1	AP^2
0.3	28.2	45.5
0.5	27.3	46.4
0.7	25.3	46.6
0.9	24.2	46.2
$\mathcal{U}(0.3, 0.9)$	**28.7**	**46.9**

(b) $\mathcal{D}$ (Adversarial Criterion)

Adv Func	AP^1	AP^2
L1	27.7	46.5
MSE	28.7	46.8
BCE	28.4	46.8
Focal	28.2	46.6
Dice	**28.7**	**46.9**

(c) Contrastive Loss Component

$\mathcal{L}_{intra}$	$\mathcal{L}_{inter}$	$\mathcal{L}_{self}$	AP^1	AP^2
✗	✗	✗	26.9	46.1
✓	✗	✗	28.1	46.6
✗	✓	✗	28.2	46.6
✓	✓	✗	28.5	46.8
✓	✓	✓	**28.7**	**46.9**

(d) Mask Perturbation

Method	Perturb	IoU^1	IoU^2
SegRefiner	Morp	58.7	74.2
	Adv	71.8	76.6
Phoenix	Morp	67.8	71.0
	Adv	**75.7**	**77.1**

(e) Noise from Model

Noisy Mask	IoU^1	IoU^2
UNet	67.6	71.6
ISNet	65.6	70.2
Adv (ours)	**75.7**	**79.3**

(f) Component Analysis

Perturb	CMRL	AP^1	AP^2	IoU^1	IoU^2
Morp	✗	22.3	45.2	65.3	69.9
Adv	✗	26.9	46.1	71.5	74.2
Morp	✓	23.8	45.5	67.8	71.0
Adv	✓	**28.7**	**46.9**	**75.7**	**77.1**

(Table 4d). Second, training quality degrades monotonically as the share of morphological noise increases (supplementary material), quantifying the realism gap. Third, Phoenix trained solely on AMP noise from LVIS generalizes zero-shot to urban-scene, medical, and semantic-segmentation domains (supplementary material), a transfer that would be unlikely if AMP merely produced semantics-agnostic synthetic noise. Notably, AMP even surpasses training directly on real UNet/ISNet errors (Table 4e), which we regard as a realism upper bound, owing to its wider error diversity and controllable magnitude.

Effect of Individual Components. Table 4f isolates the contributions of AMP and CMRL. Starting from the baseline (morphological noise with pixel-wise loss), AMP alone provides substantial improvements (+4.6% AP^1, +6.2% IoU^1), demonstrating the importance of semantic-aware noise generation. CMRL alone yields meaningful gains (+1.5% AP^1, +2.5% IoU^1), validating the impact of our tri-directional contrastive framework. Combining both components achieves the best performance (+6.4% AP^1, +10.4% IoU^1), with the combined improvement exceeding the sum of individual contributions (+8.7% IoU^1), demonstrating true synergy. This synergistic effect occurs because CMRL is more effective when encountering realistic noise patterns from AMP; the semantically meaningful error distributions enable more informative tri-directional contrastive relationships, leading to superior refinement learning. Notably, AMP provides larger marginal gains on challenging scenarios, while both components contribute substantially across all settings, confirming their complementary nature. Despite relying on randomized perturbations, Phoenix is stable across runs: over 5 independent

trainings it attains $AP^1 = 28.7 \pm 0.5$ and $AP^2 = 46.9 \pm 0.3$, with every run surpassing the strongest baseline (SAMRefiner, 21.8/45.3). A detailed stability analysis is provided in the supplementary material.

5 Conclusion

We presented Phoenix, a novel framework for mask refinement that leverages adversarial perturbation and contrastive learning to address the limitations of existing approaches. Our method generates semantically meaningful noise patterns that better reflect real-world segmentation errors and employs a tri-directional contrastive learning approach that explicitly models the relationships between ground truth, noisy input, and current prediction. Extensive experiments demonstrate that Phoenix significantly outperforms existing methods across diverse tasks and noise conditions. We believe our work offers an effective and general approach to mask refinement and opens up promising directions for future research, including self-supervised refinement and multimodal guidance integration. We further provide an extended discussion of the scope, limitations, and representative failure cases of Phoenix in the supplementary material.

Acknowledgement

This work was supported by Institute for Information & communications Technology Planning & Evaluation(IITP)grant funded by the Korea government(MSIT) (RS-2019-II190075, Artificial Intelligence Graduate School Program(KAIST))

References

1. Arnab, A., Miksik, O., Torr, P.H.: On the robustness of semantic segmentation models to adversarial attacks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 888–897 (2018) 4, 6
2. Boykov, Y.Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In: Proceedings eighth IEEE international conference on computer vision. ICCV 2001. vol. 1, pp. 105–112. IEEE (2001) 4
3. Cha, S., Yoo, Y., Moon, T., et al.: Ssul: Semantic segmentation with unknown label for exemplar-based class-incremental learning. Advances in neural information processing systems **34**, 10919–10930 (2021) 15
4. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 801–818 (2018) 3
5. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PmLR (2020) 5, 9
6. Cheng, B., Girshick, R., Dollár, P., Berg, A.C., Kirillov, A.: Boundary iou: Improving object-centric image segmentation evaluation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15334–15342 (2021) 11

7. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1290–1299 (2022) 2, 3, 12
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021) 11
9. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: International Conference on Learning Representations (ICLR) (2015) 4, 6
10. Gupta, A., Dollar, P., Girshick, R.: Lvis: A dataset for large vocabulary instance segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5356–5364 (2019) 8, 10
11. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020) 5, 9
12. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017) 12, 13
13. Huang, Z., Zhang, T.: Black-box adversarial attack with transferable model-based embedding. In: International Conference on Learning Representations (2020) 4, 6
14. Ke, L., Ye, M., Danelljan, M., Tai, Y.W., Tang, C.K., Yu, F., et al.: Segment anything in high quality. Advances in Neural Information Processing Systems **36**, 29914–29934 (2023) 12
15. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? Advances in neural information processing systems **30** (2017) 8
16. Kim, B., Jeong, J., Han, D., Hwang, S.J.: The devil is in the points: Weakly semi-supervised instance segmentation via point-guided mask representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11360–11370 (2023) 2, 4, 10, 11, 12
17. Kim, B., Shin, C., Jeong, J., Jung, H., Lee, S.Y., Chun, S., Hwang, D.H., Yu, J.: Zim: Zero-shot image matting for anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 23828–23838 (2025) 15
18. Kim, B., Yoo, Y., Rhee, C.E., Kim, J.: Beyond semantic to instance segmentation: Weakly-supervised instance segmentation via semantic knowledge transfer and self-refinement. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4278–4287 (2022) 2
19. Kim, B., Yu, J., Hwang, S.J.: Eclipse: Efficient continual learning in panoptic segmentation with visual prompt tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3346–3356 (2024) 15
20. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4015–4026 (2023) 2, 3, 5, 11
21. Kirillov, A., Wu, Y., He, K., Girshick, R.: Pointrend: Image segmentation as rendering. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9799–9808 (2020) 12
22. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. Advances in neural information processing systems **24** (2011) 4

23. Li, F., Zhang, H., Xu, H., Liu, S., Zhang, L., Ni, L.M., Shum, H.Y.: 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. pp. 3041–3050 (2023) 12
24. Li, Y., Mao, H., Girshick, R., He, K.: Exploring plain vision transformer backbones for object detection. In: European conference on computer vision. pp. 280–296. Springer (2022) 12
25. Liew, J.H., Cohen, S., Price, B., Mai, L., Feng, J.: Deep interactive thin object selection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 305–314 (2021) 11
26. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017) 12
27. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13. pp. 740–755. Springer (2014) 11
28. Lin, Y., Li, H., Shao, W., Yang, Z., Zhao, J., He, X., Luo, P., Zhang, K.: Samrefiner: Taming segment anything model for universal mask refinement. In: The Thirteenth International Conference on Learning Representations (2025) 2, 3, 4, 5
29. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015) 3
30. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017) 11
31. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (2018) 4, 6
32. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018) 9
33. Pearson, K.: Vii. note on regression and inheritance in the case of two parents. proceedings of the royal society of London **58**(347-352), 240–242 (1895) 8
34. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 724–732 (2016) 11
35. Qin, X., Dai, H., Hu, X., Fan, D.P., Shao, L., Van Gool, L.: Highly accurate dichotomous image segmentation. In: European Conference on Computer Vision. pp. 38–56. Springer (2022) 11, 12
36. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18. pp. 234–241. Springer (2015) 3, 12, 13
37. Shen, T., Zhang, Y., Qi, L., Kuen, J., Xie, X., Wu, J., Lin, Z., Jia, J.: High quality segmentation for ultra high-resolution images. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1310–1319 (2022) 12, 13
38. Tang, C., Chen, H., Li, X., Li, J., Zhang, Z., Hu, X.: Look closer to segment better: Boundary patch refinement for instance segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13926–13935 (2021) 2, 3, 4, 8

39. Tian, Z., Shen, C., Chen, H.: Conditional convolutions for instance segmentation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16. pp. 282–298. Springer (2020) 12
40. Tian, Z., Shen, C., Wang, X., Chen, H.: Boxinst: High-performance instance segmentation with box annotations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5443–5452 (2021) 2
41. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al.: Deep high-resolution representation learning for visual recognition. IEEE transactions on pattern analysis and machine intelligence **43**(10), 3349–3364 (2020) 12
42. Wang, M., Ding, H., Liew, J.H., Liu, J., Zhao, Y., Wei, Y.: Segrefiner: Towards model-agnostic segmentation refinement with discrete diffusion process. Advances in Neural Information Processing Systems **36**, 79761–79780 (2023) 2, 3, 4, 5, 8, 10
43. Wang, T., Isola, P.: Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In: International conference on machine learning. pp. 9929–9939. PMLR (2020) 5, 9
44. Wang, X., Zhang, R., Shen, C., Kong, T., Li, L.: Dense contrastive learning for self-supervised visual pre-training. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3024–3033 (2021) 5
45. Wang, Z., Li, Y., Wang, S.: Noisy boundaries: Lemon or lemonade for semi-supervised instance segmentation? In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16826–16835 (2022) 2, 4, 10, 11, 12
46. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. Advances in neural information processing systems **34**, 12077–12090 (2021) 2, 3
47. Xie, Z., Lin, Y., Zhang, Z., Cao, Y., Lin, S., Hu, H.: Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16684–16693 (2021) 5
48. Yuan, Y., Xie, J., Chen, X., Wang, J.: Segfix: Model-agnostic boundary refinement for segmentation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16. pp. 489–506. Springer (2020) 2, 3, 4, 5, 8
49. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2881–2890 (2017) 3, 12