Complementing DINO Features with Image Structure for Part Discovery

Samyak Rawlekar
University of Illinois Urbana-Champaign
Urbana, IL 61801
samyakr2@illinois.edu

Nikhil Chowdary Paleti
Independent Researcher
Sunnyvale, CA, 94087
nikhilpaleti23@gmail.com

Amey Gupta
University of Illinois Urbana-Champaign
Urbana, IL 61801
amey2@illinois.edu

Narendra Ahuja
University of Illinois Urbana-Champaign
Urbana, IL 61801
n-ahuja@illinois.edu

Abstract

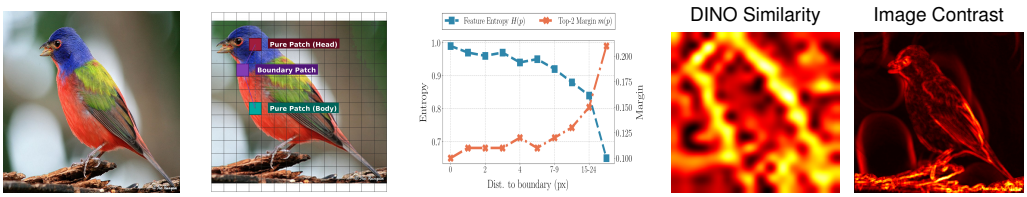
Self-supervised Vision Transformers (ViTs) like DINO encode rich semantics, enabling training-free segmentation of objects from the background and discovery of their parts via feature clustering. This paper focuses on part discovery: partitioning a foreground object into semantic parts. Existing approaches produce imprecise part boundaries. We trace this to the rigid ViT patch grid used in training DINO, where patches straddling boundaries contain pixels from multiple parts. While DINO's invariance-based training could absorb this pixel-level mixing into clean features, experiments with over 26K foreground patches show it does not: boundary-overlapping patches produce features with higher entropy and lower assignment confidence, blurring predicted part boundaries. Rather than make the massive effort of retraining DINO over geometrically coherent regions, an easier approach is to take boundaries directly from a boundary detector, which supplies the geometric information DINO's features lack. Motivated by this, we introduce Part-DINO, a training-free framework that combines boundaries from low-level image segmentation with DINO's semantic features. We use the boundaries to partition the image into regions, pool DINO features within each region, and group them to discover parts. Without any training, Part-DINO produces structurally coherent part assignments, outperforming training-free baselines across four standard benchmarks (CUB, CelebA, PartImageNet-OOD, PartImageNet-Seg) and remaining competitive with weakly-supervised and unsupervised methods.

1 Introduction

Self-supervised Vision Transformers (ViTs) encode rich object-level structure. Models like DINO (Caron et al., 2021), trained via self-distillation without any human labels, produce feature spaces that group semantic concepts. For example, on a bird, the eye patches cluster with eyes and wing patches with wings. This emergent property has enabled training-free object localization, segmentation, and coarse semantic part discovery. However, despite their strong semantic awareness, existing patch-based methods remain spatially imprecise, part assignments bleed across part contours, producing blurry boundaries between neighboring parts.

Why does a model with such semantic awareness fail to localize where one part ends and another begins? We trace this to the rigid patch grid over which features are computed. ViTs divide images into a fixed-size square patches that are agnostic to object geometry. When a patch straddles a part

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Input Image (b) Geometric Mixing (c) Feature Mixing (d) Semantic vs. Structural Edges

Figure 1: **Mixing of pixels from across part boundaries present in boundary-area patches blurs semantic boundaries.** (a) Input image. (b) A boundary patch straddling the bird's head and body pools pixels from both regions, illustrating geometric mixing in pixel space. (c) Across 26K foreground patches, feature entropy rises and the top-2 assignment margin falls as patches approach part boundaries, showing mixing in feature space, which causes ambiguity in inferences of semantic part boundaries as can be seen in (d) where edges from DINO feature similarity (left) blur at part boundaries, while edges from image contrast (right) align with them. Recovery of sharp boundaries from DINO features may thus be expected to benefit from complementary geometric information.

boundary, it pool pixels from multiple semantic regions before feature extraction (Fig 1b). While DINO's training encourages invariance across augmented views, it does not guarantee that this pixel-level mixing is resolved in the learned features. To investigate this, we analyze how patch features behave as a function of a patch's distance to the nearest ground-truth part boundary. Analyzing 26K foreground patches, we find that as patch–boundary overlap increases, feature entropy rises and assignment confidence falls (Fig 1c), confirming pixel-level mixing propagates into feature space.

This finding has a direct consequence: DINO's semantic information is useful for grouping parts, but unreliable for identifying their boundaries. As shown in Fig 1d, the edges derived from DINO feature similarity are blurred at part contours, whereas edges derived from image contrast are aligned with physical boundaries. Rather than retrain DINO over geometrically coherent regions at massive cost, an easier approach is to fuse it with an independent boundary detector.

Motivated by this, we propose Part-DINO, a training-free framework that combines the accurate boundaries from low-level image segmentation with the semantic information of DINO features. The boundaries partition the image into regions whose contours align with object geometry. We then pool DINO features within each region and use them for semantic grouping, assigning each region to its nearest part prototype.

We evaluate Part-DINO across four standard part-discovery benchmarks (CUB (Wah et al., 2011), CelebA (Liu et al., 2015), and the OOD and Seg splits of PartImageNet(He et al., 2022a)), using standard metrics (NME, NMI, and ARI). Part-DINO is training-free: it relies only on the frozen DINO features and avoids the optimization required by unsupervised methods (which train without labels) and weakly-supervised methods (which use image-level labels). Despite this constraint, Part-DINO outperforms the patch-based training-free baseline on NME and ARI across all datasets, and on NMI for three of four. The slight NMI drop on PartImageNet-OOD reflects a known metric trade-off, where ARI rewards spatial coherence over fine fragmentation (Sec.4). Furthermore, Part-DINO remains competitive with trained models, demonstrating that integrating image geometry with frozen semantic features recovers part structure without learned objectives.

Our contributions in this paper are:

1. We show that DINO's spatial imprecision stems from the ViT's patch grid. By analyzing features of patches that span boundaries, we find that pixel-level mixing propagates into the feature space, thereby blurring predicted contours.

2. We introduce Part-DINO, a training-free framework that integrates the boundary of low-level image segments to extract sharp regions and DINO features to semantically group them, achieving precise part discovery without dataset-specific training.

3. We evaluate Part-DINO across four diverse part-discovery benchmarks, demonstrating that it outperforms existing training-free baselines and is competitive with trained weakly-supervised and unsupervised methods, without any training.

2 Related Work

Part Discovery. This task aims to identify semantically consistent region (parts) across varying instances of an object category. Earlier approaches rely on weak supervision, discovering parts by using image-level category labels to optimize a classification objective that highlights discriminative regions (Huang and Li, 2020; Van Der Klis et al., 2023; Aniraj et al., 2024). Another line of work explores unsupervised methods that do not rely on any part annotations. Some approaches train a masked autoencoder that masks image patches and learns to reconstruct them (Xia et al., 2025), while others use an equivariance-based objective (Xia et al., 2024). However, both weakly and unsupervised methods need a computationally expensive and dataset-specific training phase. As an alternative, training-free approaches (Amir et al., 2021) bypass optimization entirely by clustering patch features from a frozen DINO backbone. Part-DINO follows this training-free paradigm, but use geometric regions as units instead of patches.

Self-Supervised Vision Models. DINO (Caron et al., 2021; Oquab et al., 2023) trains a Vision Transformer via self-distillation, producing patch-level embeddings that are consistent across augmented views and naturally group semantic concepts. These features encode strong spatial layouts and have been leveraged for foreground localization (Siméoni et al., 2021; Wang et al., 2023; Rawlekar et al., 2026) and coarse part discovery (Amir et al., 2021). Prior work has also investigated semantic entanglement in pretrained vision-language representations, demonstrating its impact on multi-object localization (Rawlekar et al., 2025). However, the unit used by these methods (the fixed-size square patch) is agnostic to object boundaries. Because this rigid grid ignores true object contours, boundary patches include multiple parts, producing semantically mixed features that limit spatial precision. Part-DINO avoids this by replacing the patch with geometric regions as the atomic unit, ensuring each feature vector corresponds to a single coherent part.

Region Generation. To avoid feature mixing, recent works group pixels into atomic units (Shlapentokh-Rothman et al., 2024), often relying on superpixel algorithms like SLIC (Achanta et al., 2012) to define these regions. However, while SLIC clusters pixels in a joint color-spatial space and partially respects image gradients, it imposes an explicit compactness constraint that trades boundary adherence for spatial uniformity. When part contours are weak relative to this compactness penalty, SLIC superpixels bleed across the irregular edges of natural object parts. Instead, we use the Ramp Transform and Watershed (Akbas and Ahuja, 2020, 2014). By defining the units via contrast gradients (ramp edge discontinuities), we generate regions that adhere to low-level structural boundaries.

3 Part-DINO

The Boundary Patches: We now detail the analysis behind the finding that pixel-space mixing propagates into feature space Sec.1. For each foreground patch p, we measure four quantities: (1) Euclidean distance $d(p)$ of its center to the nearest ground-truth part boundary, (2) its spatial purity $\pi(p) = |\text{majority-part pixels}|/|p_{\text{fg}}|$, (3) its feature entropy $H(p)$, computed as the entropy of the patch's softmax similarity to per-part mean features extracted from clean interior patches, and (4) its top-2 assignment margin $m(p)$, the similarity gap between p's closest and second-closest part vector.

Across $26,459$ foreground patches, smaller d corresponds to larger patch–boundary overlap. As overlap increases, it causes spatial purity ($\pi(p)$) to drop (Spearman correlation $\rho(d, \pi) = +0.71$, Supp. Fig.7a), confirming pixel-space mixing. The key finding is that this pixel space mixing propagates into the feature space: patches closer to a boundary show higher feature entropy ($\rho(d, H) = -0.38$) and a lower top-2 assignment margin ($\rho(d, m) = +0.36$). Furthermore, this boundary mixing is widespread: 21.9% of all foreground patches have a purity of $\pi < 1$. As shown in Figure 1c, H and m vary smoothly with distance d, therefore the feature quality degrades continuously as overlap increases.

This analysis has two direct implications for part discovery. First, the spatial unit must respect object geometry, since a patch that spans multiple parts cannot preserve the contour separating them. Second, the geometric signal must be independent of DINO: as Fig.1d shows, edges derived from DINO feature similarity blur at part contours, whereas edges from image contrast remain sharp and aligned with physical boundaries. Part-DINO follows from these two constraints: we extract regions from a low-level segmentation of the image without relying on DINO's semantic features(Section 3.1), build

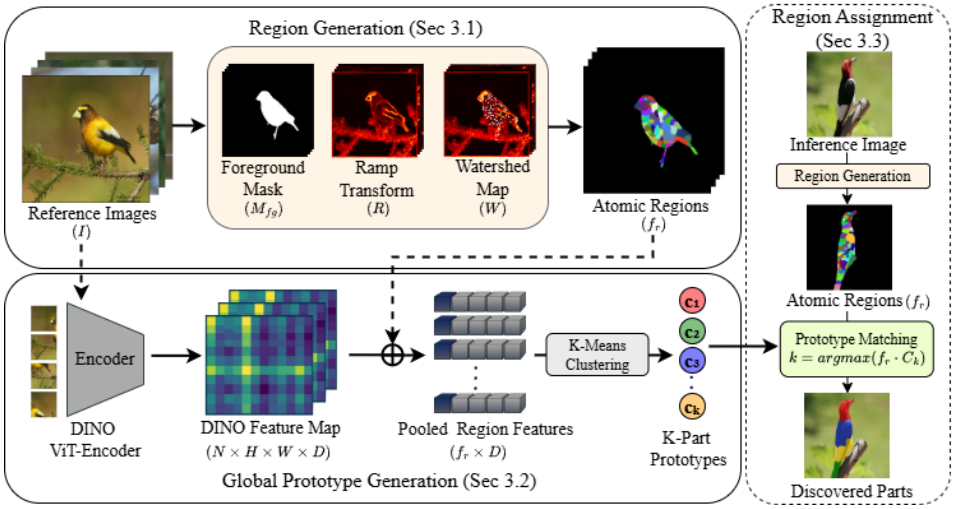


Figure 2: **Overview of Part-DINO.** Given a set of reference images $\mathcal{I}$, the Region Generation module (Sec.3.1, top) extracts a foreground mask of the object M_{fg} and performs low-level image segmentation to produce atomic regions. In parallel, the Prototype Generation module (Sec.3.2, bottom) encodes the reference images through a frozen DINO ViT to obtain dense feature maps. These semantic features are pooled within each atomic region and clustered via K-Means to establish K global part prototypes $\{c_1, c_2, \ldots, c_K\}$. At inference (right), the Region Assignment module (Sec.3.3) applies the same region generation pipeline to a test image and assigns each region to its nearest semantic prototype, producing the final part labels without any training.

Algorithm 1 Extract Region Features

Input: Image I
Output: Set of structurally bounded regions $\mathcal{R}$, Pooled region features $\mathcal{F}$

$M_{fg} \leftarrow \text{ForegroundMask}(I)$ ▷ Isolate object from background
$F \leftarrow \text{DINOv2}(I)$ ▷ Extract dense spatial feature map
$\mathcal{R} \leftarrow \text{WatershedPipeline}(I, M_{fg})$ ▷ Parse into regions
$\mathcal{F} \leftarrow \emptyset$ ▷ Initialize set for region features
for each region $r \in \mathcal{R}$ **do**
  $f_r \leftarrow \frac{1}{|r|} \sum_{p \in r} \ell_2(F_p)$ ▷ Average the features F_p of all pixels p inside r
  $\mathcal{F} \leftarrow \mathcal{F} \cup \{f_r\}$ ▷ Add pooled feature to the set
end forreturn $\mathcal{R}, \mathcal{F}$

part prototypes from these regions (Section 3.2), and assign part labels by matching region features to prototypes (Section 3.3).

3.1 Region Generation

Given an input image $I \in \mathbb{R}^{H \times W \times 3}$, our goal is to parse the object into a set of geometrically coherent regions $\mathcal{R}$ that respect part boundaries. Since we are interested in parts of the object rather than the background, we first isolate the foreground by extracting a binary mask $M_{fg} \in \{0,1\}^{H \times W}$ using foreground extraction methods. Within the foreground M, we compute a structural edge map E_{edge} from the image contrast. We apply the RAMP transform Akbas and Ahuja (2020), which estimates a per-pixel contrast magnitude based on local intensity transitions across four orientations $(0°, 45°, 90°, 135°)$. By construction, E_{edge} depends only on pixel intensities, making it independent of DINO's feature pipeline that runs in parallel to extract semantic descriptors.

To generate the atomic regions from this edge map, we apply the Watershed transform Vincent and Soille (1991) to E_{edge}, which treats the edge map as a topographic surface and floods upward from local minima, placing region boundaries along the ridges where adjacent basins meet. Seeds are placed at the local minima of E_{edge} with a minimum inter-seed distance of $d_{\min}$ pixels. Watershed tends to over-segment near weak edges, producing spurious micro-regions that lack meaningful

Algorithm 2 Part-DINO

Input: Reference set $\mathcal{D}_{ref}$, Test image I_{test}, Number of parts K, Category set $\mathcal{S}$
Output: Labeled Part Map $\mathcal{L}$

Phase 1: Global Prototype Generation

for each category $s \in \mathcal{S}$ **do**
  $\mathcal{P}^{(s)}, \mathcal{Z}^{(s)} \leftarrow \emptyset, \emptyset$
  for each image $I_i \in \mathcal{D}_{ref}^{(s)}$ **do**
    $\mathcal{R}_i, \mathcal{F}_i \leftarrow$ EXTRACT REGION FEATURES(I_i)
    $\mathcal{P}^{(s)} \leftarrow \mathcal{P}^{(s)} \cup \mathcal{F}_i; \quad \mathcal{Z}^{(s)} \leftarrow \mathcal{Z}^{(s)} \cup \{\text{DINO_CLS}(I_i)\}$
  end for
  $\mathcal{C}^{(s)} \leftarrow$ K-Means$(\mathcal{P}^{(s)}, K)$ ▷ K part prototypes
  $\bar{z}_{\text{cls}}^{(s)} \leftarrow \text{mean}(\mathcal{Z}^{(s)})$ ▷ CLS prototype
end for

Phase 2: Region Assignment

$\mathcal{R}, \mathcal{F} \leftarrow$ EXTRACT REGION FEATURES(I_{test})
$z_{\text{cls}} \leftarrow \text{DINO_CLS}(I_{test})$
$s^* \leftarrow \arg\max_{s \in \mathcal{S}} \left(z_{\text{cls}}^\top \bar{z}_{\text{cls}}^{(s)} \right)$ ▷ Category routing
$\mathcal{L} \leftarrow \emptyset$
for each region $r \in \mathcal{R}$ and feature $\hat{f}_r \in \mathcal{F}$ **do**
  $\mathcal{L}(r) \leftarrow \arg\max_k \left(\hat{f}_r^\top c_k^{(s^*)} \right)$
end forreturn $\mathcal{L}$

structure. We resolve this by merging any region under $s_{\min}$ pixels into its most similar neighbor (via DINO cosine similarity) producing the final region set $\mathcal{R}$. Since merging only removes boundaries and never places one, every boundary within the object remains a watershed ridge of E_{edge}; setting $s_{\min} = 0$ skips the merge entirely (Tab. 7).

3.2 Global Prototype Generation

Building upon the regions $\mathcal{R}$ extracted in Section 3.1, we compute a semantic representation for each such region $r \in \mathcal{R}$. We first extract the dense DINO feature map $F_{\text{patch}} \in \mathbb{R}^{H_p \times W_p \times D}$ from the input image, where D is the embedding dimension and (H_p, W_p) is the patch grid. We bilinearly upsample this to the image resolution (H, W) to get a per-pixel feature map $F \in \mathbb{R}^{H \times W \times D}$ in which each pixel p has a feature vector $F_p \in \mathbb{R}^D$. Each region's representation is then the ℓ_2-normalised average of these features over the pixels assigned to that region:

$$\hat{f}_r = \frac{1}{|r|} \sum_{p \in r} \frac{F_p}{\|F_p\|_2} \tag{1}$$

Because each region is bounded by high image contrast pixels rather than by the fixed contour of the predetermined patch grid, the pooled feature $\hat{f}_r$ aggregates pixels that lie within a single part, avoiding the loss of boundary information that compromises patch-level features.

To discover K globally consistent parts of an object category, we randomly sample a reference subset $\mathcal{D}_{\text{ref}}$ of N images from the dataset. We then perform K-Means clustering over the complete global pool of region features $(\hat{f}_r)$ from this subset to establish our category-specific semantic dictionary:

$$\mathcal{C} = c_1, \ldots, c_K = K\text{-Means}\left(\bigcup_{I_i \in \mathcal{D}_{\text{ref}}} \mathcal{F}_i; K \right) \tag{2}$$

where $\mathcal{F}_i$ is the set of region features extracted from image I_i. Each centroid $c_k \in \mathcal{C}$ acts as a global prototype for a specific part, aggregated across instances in the reference set ($\mathcal{D}_{\text{ref}}$). For datasets containing multiple object categories, this dictionary is generated independently for each class.

3.3 Region Assignment

Given a new image I_{test}, we extract its region set $(\mathcal{R})$ and the pooled features $(\hat{f}_r)$ following Section 3.1. Each region $r \in \mathcal{R}$ is then assigned to its nearest global prototype by maximizing the cosine

Table 1: Quantitative comparisons on CUB dataset across part granularities K $\in \{4, 8, 16\}$. We report NME (↓), NMI (↑), and ARI (↑) against weakly-supervised, unsupervised, and training-free methods.† indicates replicated with SAM foreground mask

Method	$K = 4$			$K = 8$			$K = 16$		
	NME ↓	NMI ↑	ARI ↑	NME ↓	NMI ↑	ARI ↑	NME ↓	NMI ↑	ARI ↑
Weakly-supervised part discovery									
Huang and Li (2020)	11.51	29.74	14.04	11.60	35.72	15.90	12.60	43.92	21.01
Van Der Klis et al. (2023)	9.12	37.82	15.26	8.52	50.08	26.96	7.64	56.87	38.05
Choudhury et al. (2021)	9.20	43.50	19.60	-	51.50	28.30	-	-	-
Van Der Klis et al. (2023) + ViT-B/14	7.70	52.59	26.66	6.34	65.01	37.90	5.95	68.63	43.41
PDiscoFormerAniraj et al. (2024)	8.19	52.88	23.22	6.23	67.59	41.35	6.44	69.54	49.99
Unsupervised part discovery									
Xia et al. (2024)	9.92	45.22	19.40	8.58	52.42	26.42	8.27	53.52	33.95
MPAE Xia et al. (2025)	7.75	58.93	33.73	5.53	68.24	45.85	5.14	79.00	66.56
Training-Free Part Discovery									
Amir et al. (2021)	-	31.18	11.21	-	47.21	19.76	-	50.57	26.14
Amir et al. (2021)†	9.92	39.77	17.94	6.26	54.03	27.18	4.87	56.92	35.10
Part-DINO (Ours)	**8.21**	**53.93**	**23.01**	**6.00**	**61.84**	**37.92**	**4.35**	**65.45**	**46.00**

Table 2: Quantitative comparison on CelebA across part granularities K $\in \{4, 8, 16\}$. We report NME (↓), NMI (↑), and ARI (↑) against weakly-supervised, unsupervised, and training-free methods. † indicates replication with SAM foreground masks.

Method	$K = 4$			$K = 8$			$K = 16$		
	NME ↓	NMI ↑	ARI ↑	NME ↓	NMI ↑	ARI ↑	NME ↓	NMI ↑	ARI ↑
Weakly-supervised part discovery									
Huang and Li (2020)	8.75	56.69	34.74	7.96	54.80	34.74	7.62	62.22	41.01
Van Der Klis et al. (2023)	11.11	75.97	69.53	9.82	62.61	51.89	9.46	77.43	70.58
Unsupervised part discovery									
GANSegHe et al. (2022b)	12.26	41.71	28.06	6.18	67.28	56.23	-	-	-
Xia et al. (2024)	12.28	50.69	34.57	10.24	58.35	46.78	-	-	-
MPAE Xia et al. (2025)	11.44	56.13	37.20	7.55	59.64	41.72	7.10	70.21	60.33
Training-Free part discovery									
Amir et al. (2021)	11.82	0.17	0.01	11.34	1.19	0.03	-	3.29	0.06
Amir et al. (2021) †	11.36	1.38	0.01	10.74	1.12	0.01	11.42	1.99	0.03
Part-DINO(Ours)	**10.81**	**45.07**	**32.20**	**10.59**	**41.53**	**28.59**	**9.15**	**41.04**	**25.20**

similarity:

$$k^* = \arg \max_{k \in \{1,\dots,K\}} \left(\hat{f}_r^\top c_k \right), \tag{3}$$

producing a dense part label map L. Since both the extracted region features and the prototypes $\mathcal{C}$ lie in the same DINO embedding space, this assignment requires no additional training. Every pixel within a region receives the same label by construction, so the final part boundaries in L coincide with the watershed boundaries computed from E_{edge}, not the rigid, low-resolution patch grid. For datasets containing multiple object categories, we maintain a separate prototype dictionary $\mathcal{C}^{(s)}$ for each category $s \in \mathcal{S}$ (Section 3.2). During assignment, we first identify the category of I_{test} by extracting its CLS token z_{cls} and compare it against per-category CLS prototypes:

$$s^* = \arg \max_{s \in \mathcal{S}} \left(z_{\text{cls}}^\top \bar{z}_{\text{cls}}^{(s)} \right), \quad \text{where } \bar{z}_{\text{cls}}^{(s)} = \frac{1}{|\mathcal{D}_{\text{ref}}^{(s)}|} \sum_{I \in \mathcal{D}_{\text{ref}}^{(s)}} \text{CLS}(I) \tag{4}$$

where both z_{cls} and $\bar{z}_{\text{cls}}^{(s)}$ are ℓ_2-normalized. Region assignment then proceeds using the selected dictionary $\mathcal{C}^{(s^*)}$. This step is itself training-free, relying solely on the CLS token's global image representation to identify the object category before part-level assignment begins.

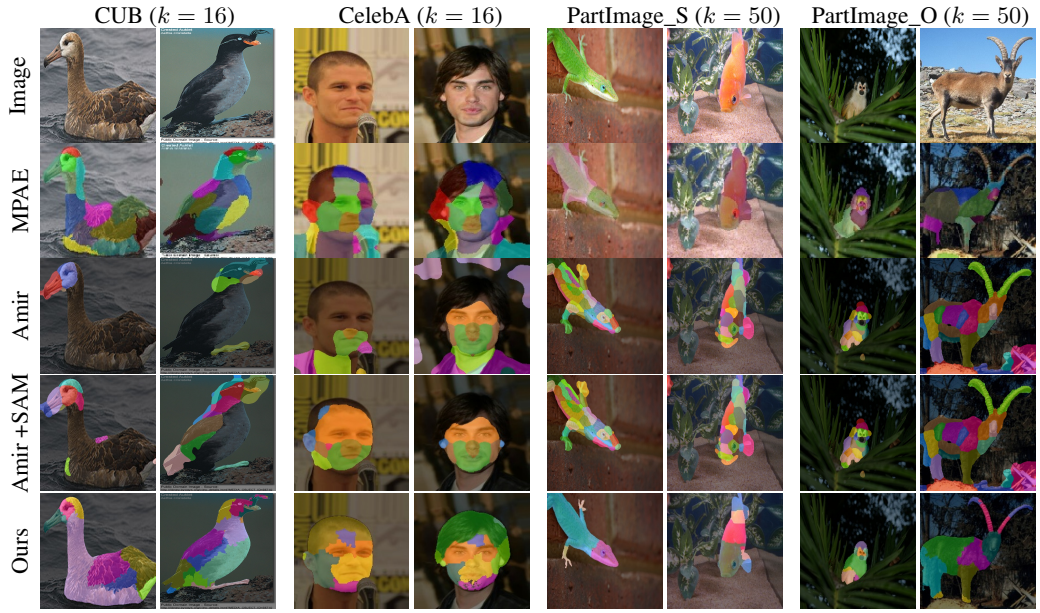


Figure 3: Qualitative comparison across CUB (K=16), CelebA (K=16), PartImageNet-S (K=50), and PartImageNet-O (K=50). MPAE is an unsupervised method that requires dataset-specific training, while Amir et al. (2021), Amir et al. (2021) + SAM, and our Part-DINO are all training-free.

4 Experiments

Dataset & Implementation Details. We evaluate Part-DINO on four widely-used benchmarks: CUB (Wah et al., 2011), CelebA (Liu et al., 2015), PartImageNet Out-of-Distribution (OOD) and PartImageNet Segmentation (He et al., 2022a). Following prior works, we use DINOv2 ViT-B/14 with registers (Darcet et al., 2023) as our frozen feature encoder, and set the input resolution to 448×448 for CUB and 224×224 for the remaining datasets. Foreground masks M are obtained using SAM3 (Carion et al., 2025). The RAMP transform is computed along four orientations ($0°, 45°, 90°, 135°$) with a maximum ramp width of 5 pixels. Watershed seeds are placed at local minima of the structural edge map with a minimum inter-seed distance of $d_{\min} = 2$ pixels, and micro-regions smaller than $s_{\min} = 10$ pixels are absorbed into their most similar DINO neighbor.

Evaluation Metrics. We evaluate Part-DINO with three standard metrics. Normalized Mutual Information (NMI) measures the mutual information between predicted cluster assignments and ground-truth labels. A high NMI indicates that predicted clusters are strongly aligned with ground-truth parts. Adjusted Rand Index (ARI) measures whether pixel pairs from the same ground-truth part are assigned the same cluster, and pairs from different parts are assigned different clusters. ARI is adjusted for chance, so a random assignment scores near zero. Normalized Mean Error (NME) is calculated as the distance between centroid of each predicted region and measure its Euclidean distance to the nearest ground-truth keypoint (landmark such as eye or beak), and report the mean of these distances normalized by image size. Lower NME implies better centroid alignment.

Results. Tables 1 and 2 show our quantitative results on the fine-grained CUB and CelebA datasets across part granularities ($K \in \{4, 8, 16\}$). On CUB, Part-DINO outperforms the patch-based training-free baseline across all K, improving NMI (56.92→65.45) and ARI (35.10→46.00) at K=16. Despite requiring no training, Part-DINO matches or exceeds several weakly-supervised and unsupervised methods (Huang and Li, 2020; Van Der Klis et al., 2023; Xia et al., 2024) at $K = 16$. On CelebA, where the patch-based baseline collapses (ARI < 0.1 across all K), Part-DINO outperforms the unsupervised GANSeg at $k = 4$ (ARI: 32.20 vs. 28.06), though trained methods remain stronger. Table 5 shows results on the PartImageNet OOD (PartImageNet$_O$) and Segmentation (PartImageNet$_S$) splits. On PartImageNet$_O$, Part-DINO achieves consistently higher ARI but slightly lower NMI compared to patch-based baselines. This reflects a known trade-off: NMI favors over-segmentation into many small, pure clusters, while ARI penalizes such fragmentation. By grouping spatially coherent regions, Part-DINO prioritizes structurally consistent parts, leading to higher ARI at the cost

Table 3: Impact of the atomic unit on CUB ($k = 4$). Combining region-based assignment with region-based prototypes (row 4) yields the largest improvement across all metrics.

Unit	Prototypes	NME (↓)	NMI (↑)	ARI (↑)
Patches	Patch-based	9.22	32.83	13.96
Patches	Region-based	8.30	39.33	16.10
Regions	Patch-based	9.02	42.80	18.13
Regions	Region-based	8.21	53.93	23.01

Table 4: Region generation methods on CUB ($k = 16$). Even at coarser granularity, RAMP + Watershed outperforms fine-grained SLIC.

Method	Regions	NME ↓	NMI ↑	ARI ↑
Patches	–	4.85	49.03	22.54
SLIC	250	5.00	58.55	36.94
	500	4.99	60.78	38.64
RAMP + Watershed	142	4.44	61.96	43.13
	500	4.35	65.45	46.00

Table 5: Quantitative comparison on PartImageNet Out-of-Distribution (PartImageNet$_O$) and Segmentation (PartImageNet$_S$) splits across $K \in \{8, 25, 50\}$. Part-DINO shows consistent improvements over the training-free baseline on PartImageNet$_S$ across all granularities and metrics, and remains competitive on PartImageNet$_O$. † indicates replicated with SAM foreground mask

Method	PartImage_O (%)						PartImage_S (%)					
	$K = 8$		$K = 25$		$K = 50$		$K = 8$		$K = 25$		$K = 50$	
	NMI ↑	ARI ↑	NMI ↑	ARI ↑	NMI ↑	ARI ↑	NMI ↑	ARI ↑	NMI ↑	ARI ↑	NMI ↑	ARI ↑
Weakly-supervised part discovery												
Huang and Li (2020)	5.88	1.53	7.56	1.25	10.19	1.05	-	-	-	-	-	-
Van Der Klis et al. (2023)	27.13	8.76	32.41	10.69	41.49	14.17	10.68	32.43	18.44	40.70	21.98	39.40
Van Der Klis et al. (2023)(ViT-B/14)	19.28	34.72	28.23	25.93	29.48	28.68	16.68	32.01	46.81	40.70	34.13	39.40
Aniraj et al. (2024) (ViT-B/14*)	28.84	55.66	43.36	62.82	44.48	57.91	-	-	-	-	-	-
Aniraj et al. (2024)(ViT-B/14†)	29.00	52.40	44.71	59.27	46.29	62.21	20.29	38.90	43.18	64.61	44.06	60.10
Unsupervised part discovery												
Xia et al. (2024)	27.85	40.77	-	-	-	-	25.17	37.50	-	-	-	-
MPAEXia et al. (2025)	32.90	66.17	39.28	68.87	53.65	74.22	33.51	65.05	37.22	68.04	55.10	73.52
Training-Free part discovery												
Amir et al. (2021)	19.17	7.59	31.46	14.16	37.81	16.50	-	-	-	-	-	-
Amir et al. (2021) †	**32.86**	54.77	**37.19**	57.87	**40.41**	59.99	25.59	57.69	27.08	62.92	32.12	64.73
Part-DINO(Ours)	30.07	**63.27**	31.14	**64.93**	30.13	**65.42**	**37.21**	**66.10**	**41.54**	**66.94**	**41.80**	**66.97**

of marginally lower NMI. On PartImageNet$_S$, Part-DINO improves over the training-free baseline on both NMI and ARI for K $\in \{8, 25, 50\}$, maintains an ARI above 66, and outperforms the fully trained SOTA MPAE at K = 8 on both metrics (37.21/66.10 vs. 33.51/65.05) despite no training.

5 Analysis

Unit (Patches vs Regions): To isolate the impact of the atomic unit, we test all combinations of patches and regions for prototype construction and label assignment. As shown in Table 3, improving only the prototype quality while keeping patches as the assignment unit (row 2) yields marginal gains over the baseline (+6.50 NMI, +2.14 ARI). This is expected: while the target prototypes are cleaner, the patch-based assignment still blurs features across part boundaries. Improving only the assignment unit while keeping patch-based prototypes (row 3) yields a larger gain (+9.97 NMI, +4.17 ARI), as geometric region produces clean query features even when the targets are noisy. Combining both changes (row 4), we match region features with region prototypes, reducing mixing on both sides of the assignment and improving NMI (32.83 $\rightarrow$ 53.93) and ARI (13.96 $\rightarrow$ 23.01).

Region Generation: We compare RAMP-based Watershed Akbas and Ahuja (2020, 2014) against SLIC Achanta et al. (2012), keeping all other components fixed. As shown in Table 4, both methods improve over the patch baseline. At the same granularity of 500 regions, RAMP + Watershed improves ARI by +7.36 over SLIC, and even at a coarser granularity (142 regions) outperforms fine-grained SLIC (500 regions). The difference is visible in Figure 5b: SLIC impose an explicit compactness constraint that trades boundary adherence for spatial uniformity. RAMP + Watershed places boundaries along intensity gradients, resulting in regions that respect object geometry.

Region Granularity. To study the effects of region granularity, we vary the minimum inter-seed distance $d_{\min}$. As shown in Figure 4, finer regions improve part discovery. Decreasing $d_{\min}$ from 15 to 2 increases the average number of regions from 22 to 500, improving all metrics (e.g ARI 25.21 $\rightarrow$ 46.00). The performance drop with coarser regions occurs because large regions span multiple

Table 6: Structural vs. Semantic edge maps on CUB. Regions derived from purely geometric edges consistently outperform those derived from DINOv2 semantic edges across all K, with the gap widening at finer granularities (e.g., +9.06 NMI and +7.05 ARI at K=16).

K	Edges	NME (↓)	NMI (↑)	ARI (↑)
4	Semantic	8.23	48.73	19.84
	Geometry	8.21	53.93	23.01
8	Semantic	6.11	50.91	24.60
	Geometry	6.00	61.84	37.92
16	Semantic	4.63	56.39	38.95
	Geometry	4.35	65.45	46.00

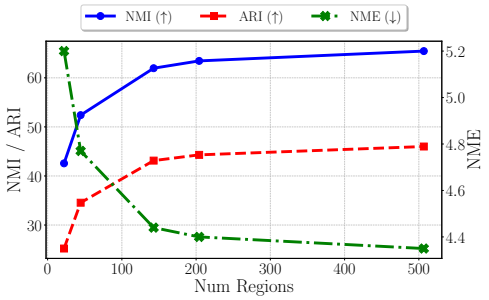
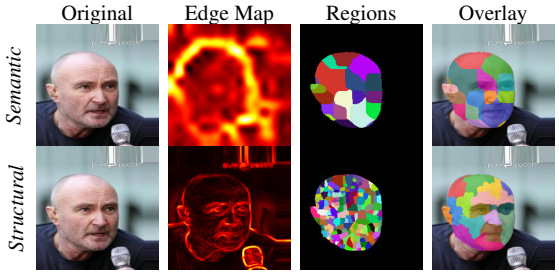
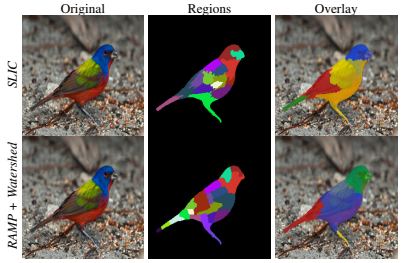


Figure 4: Effect of region granularity on CUB (K=16). Finer granularities improve part discovery across all metrics, with performance saturating around 300 regions.



(a) Structural vs. Semantic Edge Map.



(b) Watershed vs. SLIC.

Figure 5: Qualitative experiments on region generation. (a) The structural edges from RAMP align with object contours, while DINO's semantic edges are blurred. (b) Watershed regions are separable (head, body, wing, and tail), while SLIC prioritizes spatial compactness and cuts across them.

parts, pooling features across part contours. In contrast, finer regions ensure that each unit remains contained within a single part, thus preserving the boundary precision.

Structural vs. Semantic Edge Map. Our region generation relies on structural edge map (E_{edge}) derived from the RAMP transform, explicitly avoiding the use of DINO semantic boundaries. As shown in Table 6, compared to a baseline semantic similarity between patch features, structural edges consistently perform better across all granularities. The performance the gap widening as K increases, at $K = 16$, structural edges improve NMI by +9.06 and ARI by +7.05. Figure 5a shows semantic edges are blurred at object boundaries, producing regions that bleed across physical contours.

Computational Efficiency. We compare the computational efficiency of our method with (Amir et al., 2021). Both share an identical DINOv2 backbone (~40 ms) and SAM3 masking (~560 ms). The main difference lies in Part-DINO's region-construction pipeline: RAMP (32 ms), watershed (25 ms), merging (6 ms), and pooling (1 ms), which introduces a negligible total overhead of 72 ms per image. For this minimal cost, our method results in +8.6 NMI and +10.9 ARI at $d_{\min} = 2$. More detailed comparison of each component across different d_{min} is shown in Appendix Sec.A and Tab.8.

6 Discussion

We have presented Part-DINO, a training-free framework motivated by an empirical finding: pixel-space mixing at boundary patches propagates into feature space. Across 26k foreground patches, feature entropy rises and the top-2 assignment margin falls as patch–boundary overlap increases, making the ViT patch the bottleneck for spatially precise part discovery. Part-DINO addresses this by replacing the patch with regions whose boundaries come from low-level image segmentation alone, independent of DINO. We pool DINO features within each region and use them only to group regions into parts, not to place boundaries. Part-DINO outperforms the patch-based training-free

baseline on NME and ARI across all four benchmarks and granularities, and matches or exceeds several training-based methods.

We present **two limitations**. (1) Fixed part granularity. Part-DINO requires the number of parts (K) to be specified in advance. While this is a common assumption across all part discovery methods, it limits applicability to settings where the correct granularity is unknown or varies across object categories. (2) Dependence on contrast. Because our region boundaries are derived from the RAMP transform, which relies on local intensity contrast, the edge response weakens in low-contrast scenes, uniform lighting or minimal texture variation between parts, leading to under-segmentation. Conversely, strong illumination gradients or cast shadows can introduce spurious edges that do not correspond to part boundaries.

Future work. Part-DINO discovers parts as unlabeled clusters. A natural extension is to assign semantic labels to these discovered regions by leveraging vision-language models such as CLIP, bridging part discovery and open-vocabulary part segmentation. **Broader Impact.** Positively, our training-free approach reduces the computational energy footprint typically required for dataset-specific part discovery. Negatively, enhanced unsupervised localization capabilities could be intentionally misused for unauthorized biometric surveillance.

Acknowledgments

This work was supported in part by the Office of Naval Research under Grant N00014-24-1-2169, the IBM-Illinois Discovery Accelerator Institute (IIDAI), and the Amazon–Illinois Center on AI for Interactive Conversational Experiences. This research also utilized resources from the NCSA Delta system, supported by the National Science Foundation under Award OAC-2005572.

References

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

Emre Akbas and Narendra Ahuja. Low-level hierarchical multiscale segmentation statistics of natural images. *IEEE transactions on pattern analysis and machine intelligence*, 36(9):1900–1906, 2014.

Emre Akbas and Narendra Ahuja. Low-level multiscale image segmentation and a benchmark for its evaluation. *Computer Vision and Image Understanding*, 199:103026, 2020.

Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2(3):4, 2021.

Ananthu Aniraj, Cassio F Dantas, Dino Ienco, and Diego Marcos. Pdiscoformer: Relaxing part discovery constraints with vision transformers. In *European Conference on Computer Vision*, pages 256–272. Springer, 2024.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *CVPR*, pages 9650–9660, 2021.

Subhabrata Choudhury, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Unsupervised part discovery from contrastive reconstruction. *Advances in Neural Information Processing Systems*, 34:28104–28118, 2021.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023.

Ju He, Shuo Yang, Shaokang Yang, Adam Kortylewski, Xiaoding Yuan, Jie-Neng Chen, Shuai Liu, Cheng Yang, Qihang Yu, and Alan Yuille. Partimagenet: A large, high-quality dataset of parts. In *European Conference on Computer Vision*, pages 128–145. Springer, 2022a.

Xingzhe He, Bastian Wandt, and Helge Rhodin. Ganseg: Learning to segment by unsupervised hierarchical image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1225–1235, 2022b.

Zixuan Huang and Yin Li. Interpretable and accurate fine-grained recognition via region grouping. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8662–8672, 2020.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Samyak Rawlekar, Yujun Cai, Yiwei Wang, Ming-Hsuan Yang, and Narendra Ahuja. Efficiently disentangling clip for multi-object perception. *arXiv preprint arXiv:2502.02977*, 2025.

Samyak Rawlekar, Amitabh Swain, Yujun Cai, Yiwei Wang, Ming-Hsuan Yang, and Narendra Ahuja. Finding distributed object-centric properties in self-supervised transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 31326–31335, 2026.

Michal Shlapentokh-Rothman, Ansel Blume, Yao Xiao, Yuqun Wu, Sethuraman Tv, Heyi Tao, Jae Yong Lee, Wilfredo Torres, Yu-Xiong Wang, and Derek Hoiem. Region-based representations revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17107–17116, 2024.

Oriane Siméoni, Gilles Puy, Huy V Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. *arXiv preprint arXiv:2109.14279*, 2021.

Robert Van Der Klis, Stephan Alaniz, Massimiliano Mancini, Cassio F Dantas, Dino Ienco, Zeynep Akata, and Diego Marcos. Pdisconet: Semantically consistent part discovery for fine-grained recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1866–1876, 2023.

Luc Vincent and Pierre Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(06):583–598, 1991.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Yangtao Wang, Xi Shen, Yuan Yuan, Yuming Du, Maomao Li, Shell Xu Hu, James L Crowley, and Dominique Vaufreydaz. Tokencut: Segmenting objects in images and videos with self-supervised transformer and normalized cut. *IEEE TPAMI*, 45(12):15790–15801, 2023.

Jiahao Xia, Wenjian Huang, Min Xu, Jianguo Zhang, Haimin Zhang, Ziyu Sheng, and Dong Xu. Unsupervised part discovery via dual representation alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10597–10613, 2024.

Jiahao Xia, Yike Wu, Wenjian Huang, Jianguo Zhang, and Jian Zhang. Unsupervised part discovery via descriptor-based masked image restoration with optimized constraints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8668–8677, 2025.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction clearly state the contributions made in the paper and important assumptions and limitations. Our experimental results justify our claims

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: Yes, we have provided a limitation section in the paper.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A].

Justification: [N/A].

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: See implementation details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes].

Justification: We provide the code details (implementation details) to faithfully reproduce the main experimental results.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: See implementation details section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No].

   Justification: We follow the same experimentation of prior works. Given the use of or centroids, the results will be exactly same.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: The necessary values are presented in the implementation details, results sections and the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We reviewed the NeurIPS Code of Ethics.584

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: Provided in Sec 6. Discussion

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification:[N/A].

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: We have properly credited and cited the original owners of assets where appropriate.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: We provide the necessary details of the code in the paper. We do not have any other assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: [N/A].

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A].

Justification: [N/A].

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

Appendix

In this Appendix, we provide an extended analysis of the computational efficiency of our framework (Appendix A), further empirical validation of the geometric mixing that corrupts ViT boundary patches (Appendix B), and extended qualitative comparisons evaluating structural versus semantic boundaries (Appendix C) as well as Watershed versus SLIC region generation across datasets (Appendix D). Lastly, we demonstrate that region generation remains effective even without the DINO-based micro-region merge (Appendix E).

A Computational Efficiency: Extended Analysis

Setup. We report means over the CUB test split at resolution 448×448 on a single NVIDIA A40 GPU with 48 CPU cores (no batching). The (Amir et al., 2021) patch baseline shares the identical backbone and SAM3 segmenter call, only the feature-extraction and assignment stages differ.

Stage-level breakdown. Table 8 decomposes latency into seven stages. The *shared backbone* (DINO forward + SAM3 mask) accounts for approximately 600 ms in both methods and represents the dominant cost. Among Part-DINO's method-specific stages, RAMP edge estimation and watershed seeding total ~57 ms and are independent of $d_{\min}$. The merge stage scales mildly with the number of atomic regions: at $d_{\min} = 2$ (538 regions, mean), merge takes 7.7 ms; at $d_{\min} = 15$ (23 regions), it takes 5.2 ms. Region pooling via `torch.scatter_add` requires ~1 ms regardless of granularity.

B Boundary Analysis Details

In the main text, we hypothesize that the fixed-size square grid (14×14 pixels) used by Vision Transformers acts as a fundamental bottleneck for spatial precision. Figure 7 visualizes the empirical validation of this hypothesis through an analysis of $N = 26,459$ foreground patches from the PartImageNet dataset, examining how semantic representation degrades as a function of distance to physical object contours.

To rigorously evaluate this geometric relationship, we first constructed ground-truth part centroids using high-purity patches ($\pi > 0.95$) from the training set, anchoring them to isolated, unambiguous semantic parts. For each patch p, we calculated its Euclidean distance d to the nearest ground-truth part boundary. We then mapped this distance against three key patch-level metrics:

1. Spatial Purity (π): The fraction of the patch's foreground pixels belonging to its majority part label.
2. Feature Entropy (H): The Shannon entropy of the patch's feature distances relative to the part-anchored centroids, measuring representational ambiguity.
3. Top-2 Margin (m): The difference in similarity between the nearest and second-nearest part centroids, serving as a measure of assignment confidence.

As illustrated by the distance curves in Figure 7, a patch's proximity to a boundary heavily dictates its semantic reliability. We observe a strong positive correlation ($\rho(d, \pi) = +0.71$) between boundary distance and spatial purity; patches directly on or adjacent to a contour unavoidably pool pixels from distinct parts.

Crucially, this geometric mixing corrupts the underlying feature space. As patches move closer to the boundary (decreasing d), feature entropy rises ($\rho(d, H) = -0.38$) and the assignment margin steadily narrows ($\rho(d, m) = +0.36$). While the dataset-wide mean patch purity is high (0.952), the sharp degradation of these metrics at $d \approx 0$ quantitatively demonstrates that square ViT patches become inherently ambiguous exactly where precise boundary localization is required.

C Extended Qualitative Analysis: Semantic vs. Structural Boundaries

In the main text, we established that extracting boundaries via semantic feature similarity leads to spatial imprecision, whereas relying on structural image contrast preserves object contours. Figures

6, 8, and 9 provide an extended qualitative comparison of these edge map modalities across various datasets.

As demonstrated in the figures, edges derived from DINO similarity inherently suffer from feature degradation near part transitions, causing the predicted regions to bleed across actual object boundaries. By contrast, our structural boundary formulation leverages the RAMP transform to isolate edge discontinuities independent of the DINO feature space. Across CelebA (Figure 6), and other complex object categories (Figures 8 and 9), the structural boundaries consistently reduce semantic bleeding artifacts and produce sharp parts.

D Extended Qualitative Analysis on Region Generation: SLIC vs. Watershed

To construct atomic regions that do not arbitrarily cross object parts, the choice of the grouping algorithm is critical.Figures 10 and 11 provide an extended ablation comparing our chosen Watershed transform against SLIC superpixels across a variety of complex categories.

While SLIC clusters pixels based on a joint color-spatial space, it imposes a strict spatial compactness penalty. As seen in the SLIC rows of Figures 10 and 11, when structural contours are weak or highly irregular, this compactness constraint dominates, forcing arbitrary, grid-like boundaries that cut directly across anatomical shapes. Conversely, our Watershed approach operates on the topographic valleys of the structural gradient. As shown in the Watershed rows, this allows the generated atomic regions to break free from spatial rigidity and seamlessly wrap around the true, natural contours of the physical objects.

E DINO-Free Region Generation

DINO enters region generation only through the micro-region merge (Sec. 3.1), which absorbs each watershed region smaller than $s_{\min}$ pixels into its most similar neighbor. Merging only removes boundaries and never places one. DINO therefore decides which watershed regions are grouped, the same role it plays in region assignment (Sec. 3.3), while boundary placement depends on E_{edge} alone. To confirm that the gains do not depend on this step, we disable it ($s_{\min} = 0$), making region generation fully independent of DINO.

Table 7 reports the result on CUB at $K = 16$. Disabling the merge adds 18.5 regions per image and changes NMI and NME by only 0.03, while ARI drops by 1.91. Even without the merge, Part-DINO outperforms the patch-based baseline by +8.09 NMI and +8.82 ARI, with 0.51 lower NME. The gains therefore come from the DINO-independent boundaries, not from the merge.

Table 7: Effect of the DINO-based micro-region merge on CUB ($K = 16$). With $s_{\min} = 0$, region generation uses no DINO features. Both Part-DINO rows come from the same run. † replicated with SAM foreground mask.

Method	Regions/img	NME ↓	NMI ↑	ARI ↑
Amir et al. (2021)†	–	4.87	56.92	35.10
Part-DINO ($s_{\min} = 10$, default, rerun)	506.7	4.33	64.98	45.83
Part-DINO ($s_{\min} = 0$, no merge)	525.2	4.36	65.01	43.92

Table 8: Timing breakdown (ms), average regions, and clustering performance metrics.

	Runtime Breakdown (ms)									Metrics		
Method	DINO	SAM	RAMP	Wshed	Merge	Pool	Assign	Total	Avg. Reg.	NME (↓)	NMI (↑)	ARI (↑)
(Amir et al., 2021) (patch)	40.0	552.4	0.0	0.0	0.0	6.0	1.1	599.5	1024	4.87	56.92	35.10
Part-DINO ($d_{min} = 15$)	40.2	558.3	32.4	23.8	5.2	1.3	0.4	664.9	23.1	5.20	42.57	25.21
Part-DINO ($d_{min} = 5$)	39.7	562.9	32.8	24.5	6.1	1.3	1.4	672.2	151.1	4.44	61.96	43.13
Part-DINO ($d_{min} = 2$)	39.6	563.1	31.9	29.0	7.7	1.4	3.7	680.2	562.1	4.35	65.45	46.00

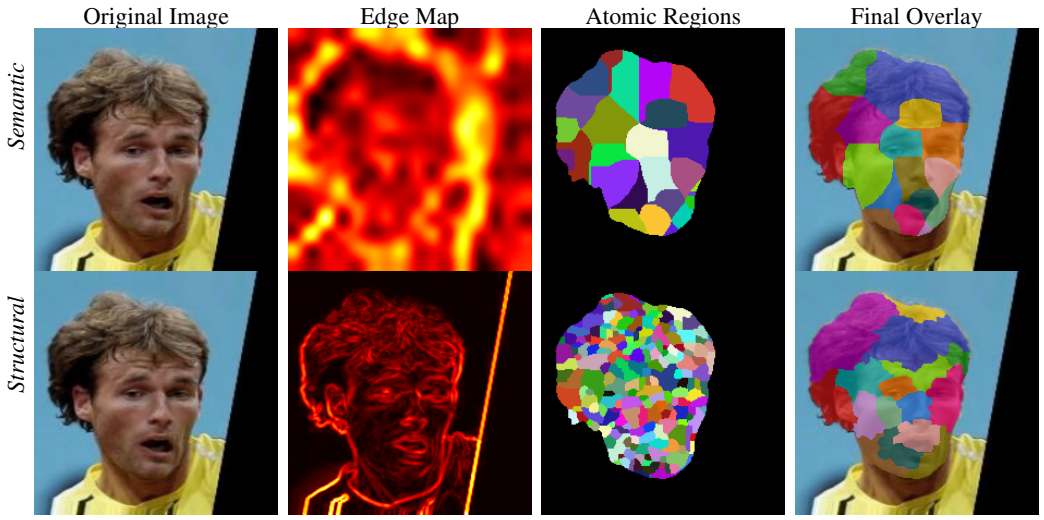


Figure 6: **Extended Qualitative Analysis (CelebA).** Comparison of edge map modalities. Structural boundaries consistently eliminate the semantic bleeding artifacts present in the patch-based baseline.

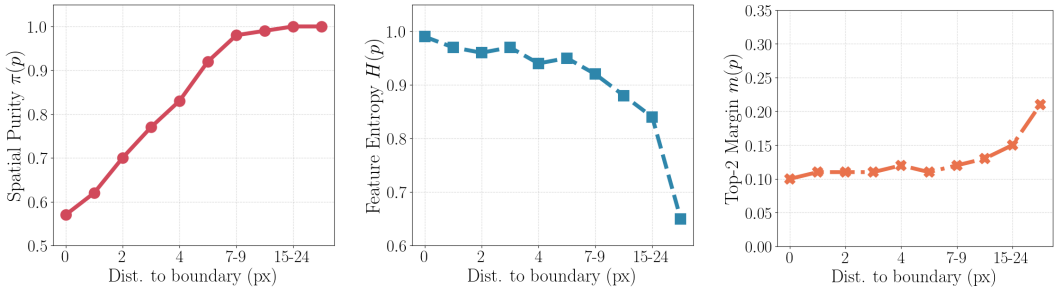


Figure 7: **Quantifying the boundary bottleneck.** Across 26,459 foreground patches, all three reliability metrics correlate monotonically with distance to the nearest part boundary: spatial purity rises $\rho(d, \pi) = +0.71$), while feature entropy falls (($\rho(d, H) = -0.38$), and top-2 assignment margin grows ($\rho(d, m)$ = +0.36), confirming that geometric mixing propagates into feature space

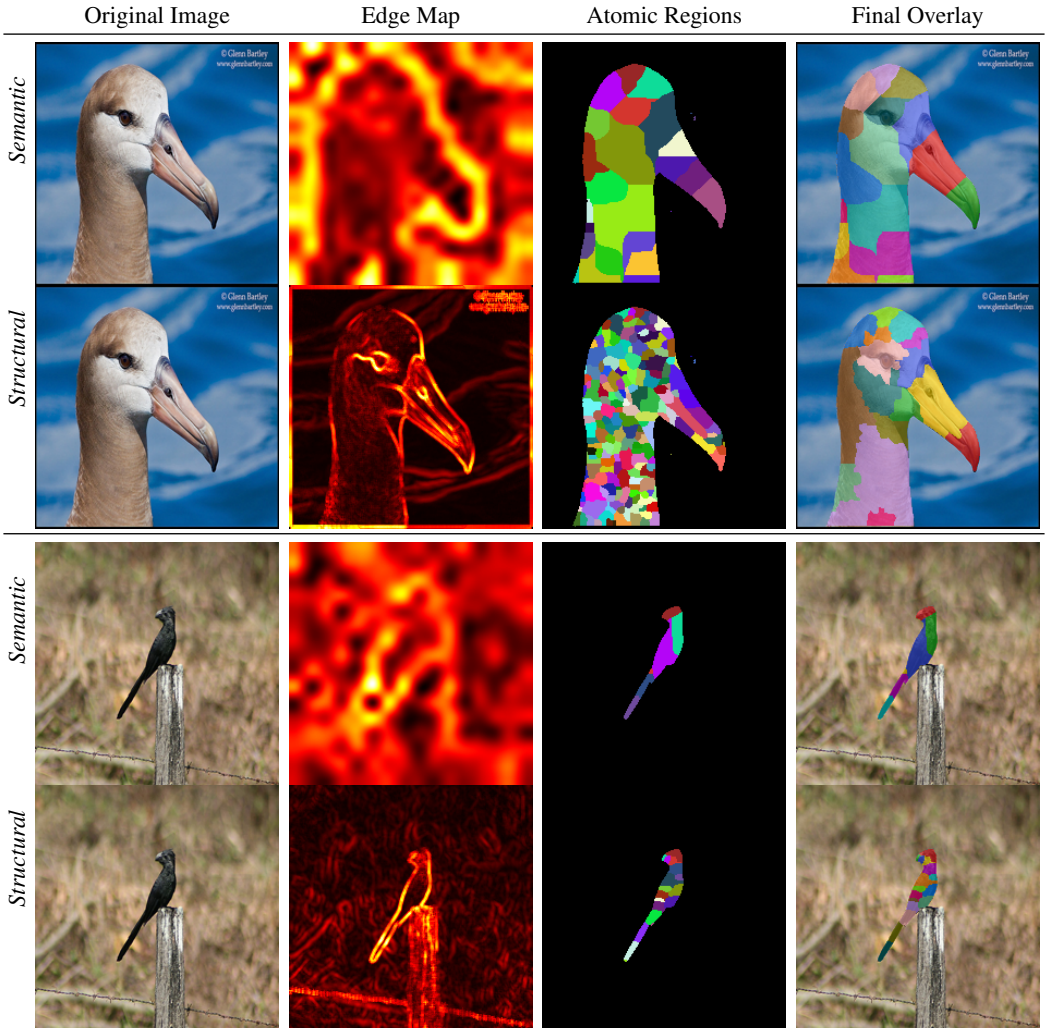


Figure 8: **Extended Qualitative Analysis (CUB).** Comparison of edge map modalities on fine-grained bird categories, highlighting precision improvements in part boundaries.

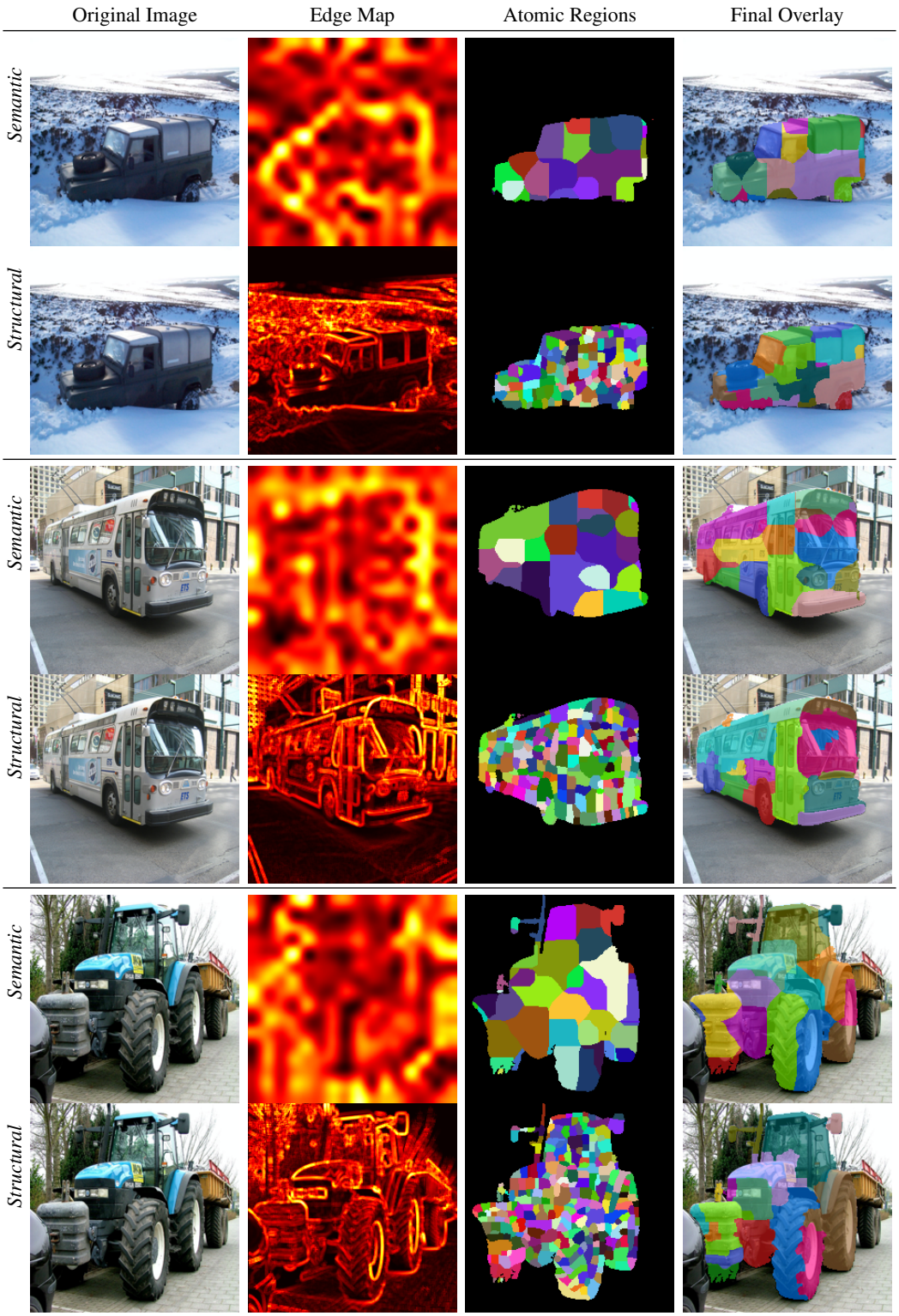


Figure 9: **Extended Qualitative Analysis (PartImageNet).** Comparison of structural vs. semantic edge modalities on Out-of-Distribution (OOD) and Segmentation (Seg) examples. Structural boundaries demonstrate robust geometric alignment regardless of semantic category.

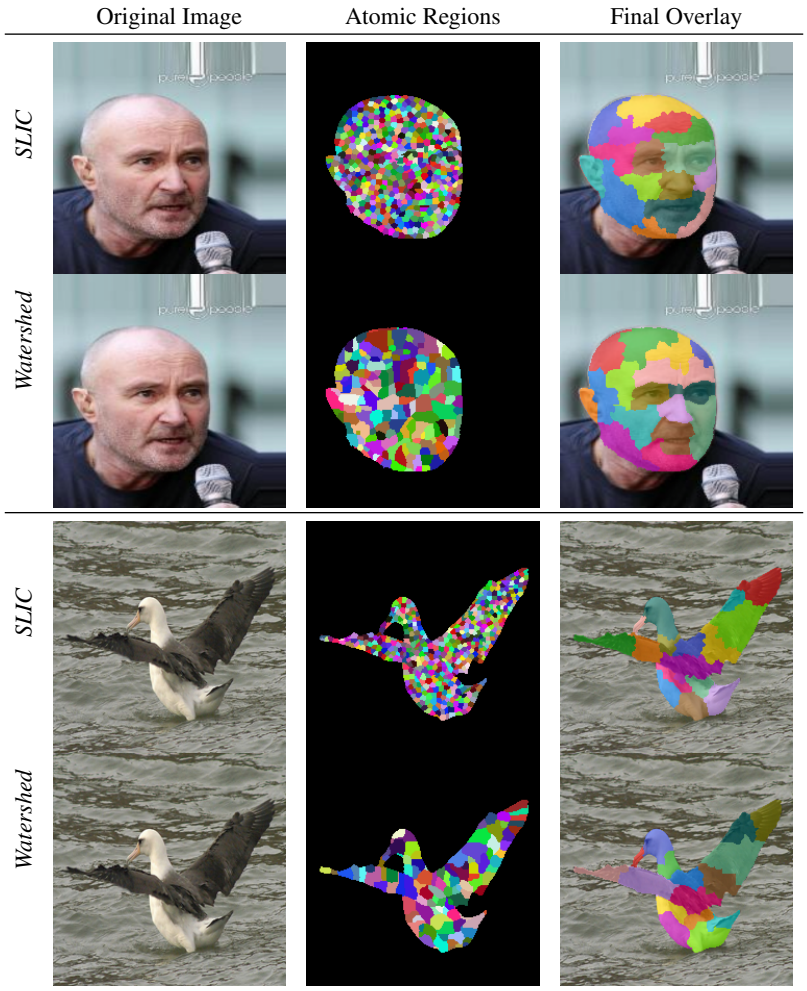


Figure 10: **Extended Ablation on Region Generation (Part 1).** SLIC enforces arbitrary spatial compactness, forcing grid-like boundaries that ignore true anatomical shapes. In contrast, our Watershed approach naturally follows the topographic valleys of the structural gradient, seamlessly wrapping around physical object contours.

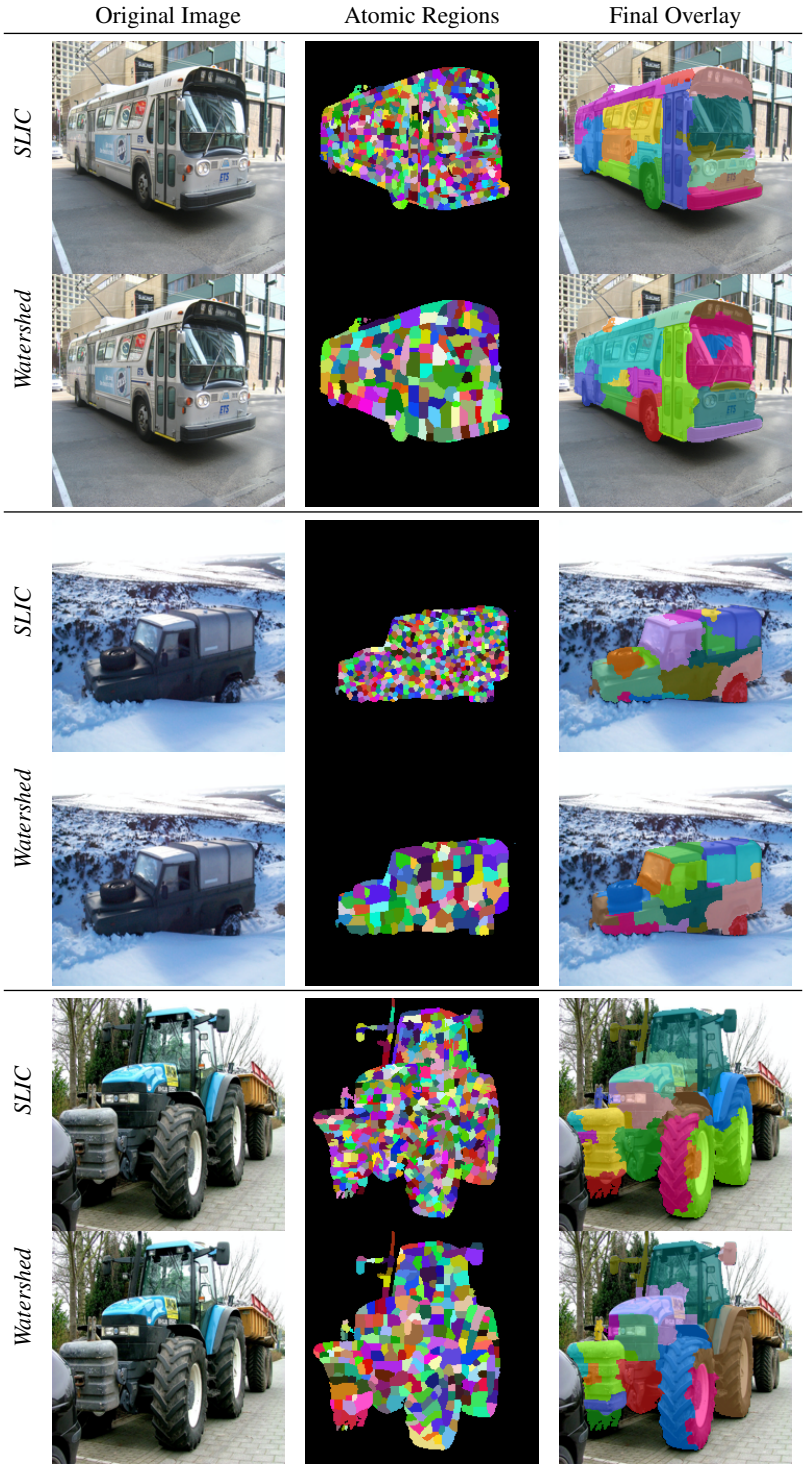


Figure 11: **Extended Ablation on Region Generation (Part 2).** Continued comparison across complex PartImageNet categories, demonstrating the robustness of Watershed boundaries against rigid SLIC superpixels.