RATS! Patches Talk Through Registers: Emergent Parts in Register Attention Transformers

Timing Yang[1] **Predrag Neskovic**[2] **Jansen Seheult**[3] **Wenchao Han**[3]
Anand Bhattad[1] **Alan Yuille**[1] **Feng Wang**[1*]

[1]Johns Hopkins University [2]Office of Naval Research, Arlington, VA
[3]Department of Laboratory Medicine and Pathology, Mayo Clinic, MN, USA

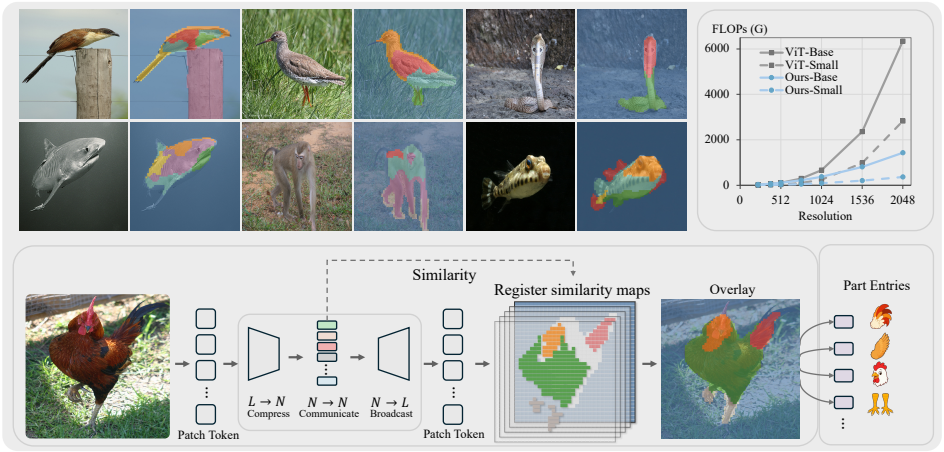


Figure 1: ***RATS*** discovers compositional part structure through a register-token bottleneck. *Top:* unsupervised part segmentations emerge across diverse categories (birds, snake, shark, monkey, fish). *Top-right:* the bottleneck makes inference scale efficiently with image resolution. *Bottom:* each block routes patch–patch communication through N register tokens via a three-step *compress–communicate–broadcast* attention; per-register similarity maps over the patch grid recover semantic parts, which we catalog into a small dictionary of reusable *part entries*.

Abstract

When humans see a bird, they recognize far more than just "bird" — they see a head, wings, and talons, a structured assembly of reusable parts that can be identified across every bird they have ever seen. We ask whether a self-supervised visual model can discover the same compositional structure on its own. To this end, we propose **RATS** (**R**egister **A**ttention **T**ransformer**s**), which decomposes the classification token into N learnable register tokens that route the information of L patches through an $L{\to}N{\to}N{\to}L$ bottleneck via a three-step *compress–communicate–broadcast* attention. The N registers are partitioned across the H attention heads, so that registers assigned to different heads do not interact with each other. Without auxiliary losses or part annotations, each register spontaneously specializes into a proto-semantic region whose emerging structure resembles object parts. RATS surpasses all baselines by +9.8 mIoU on average across five segmentation benchmarks at a matched token budget, with consistent gains on ADE20K (+1.11 mIoU) and COCO (+0.2 AP^m). Its register dictionary further exhibits part-level consistency and semantic proximity across related categories. Our results suggest that RATS may provide a useful architectural prior for structured and interpretable visual representation learning.

*Corresponding author. Email: `wangf3014@gmail.com`. Code: `https://github.com/yangtiming/RATS`.

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

1 Introduction

When a person looks at a photograph of a bird, they do not merely recognize "bird." They see a head, two wings, talons gripping a branch — a structured assembly of parts, each meaningful on its own and recognizable across every bird they have ever seen. This decomposition is immediate and effortless: humans naturally parse visual scenes into reusable local structures, whether a knife's metal blade and wooden handle, or a car's tires, body, and windows. These parts are *transferable* across instances (all cats have ears), *shared* across categories (bicycle tires and car tires are fundamentally similar), and *composable* (different parts combine into different objects). They are, in a precise sense, the atomic vocabulary of visual understanding — and, we argue, the representations that a visual model should discover on its own.

Self-supervised learning has produced surprisingly powerful visual representations. Methods ranging from masked patch prediction [32] and cross-view consistency [13] to self-distillation [11] have shown that rich structure can be learned from images alone, without any labels. The DINO family in particular has pushed this further: DINOv2 [52] features, when visualized via PCA, reveal striking cross-image correspondences—the same principal component highlights cat ears, dog ears, and rabbit ears simultaneously, or maps bicycle wheels onto car wheels across entirely different images. Part-level structure, it turns out, is already latent in self-supervised features. Yet surfacing it as an explicit, structured representation remains an open problem. Standard ViTs have no mechanism to do so: fully-connected self-attention collapses everything into a single global classification token(`[CLS]`), with no internal token ever asked to represent a specific region or part. The patch tokens, while individually informative, are never grouped into coherent regions—the model has no representational commitment to parts. This suggests a natural next step: rather than treating an image as a flat bag of patch tokens to be globally summarized, can we *group* these tokens into meaningful region-level representations—letting the model explicitly organize what it has implicitly learned?

Vision Transformers provide no explicit grouping mechanism, and several lines of work address this by introducing intermediate latent tokens as grouping containers. Superpixel-based methods [48] follow image boundaries but are restricted to local context, lacking the global information needed for part identity. Object-centric methods [45, 55, 24] let slot vectors compete to represent scene elements, but reconstruction-driven supervision yields object-level masks rather than part-level semantics, and hard mutual exclusion breaks semantic coherence. GroupViT [69] achieves semantic grouping but requires image-text pairs, specializing tokens to text-described categories rather than structural parts. Perceiver IO [38] compresses inputs into a compact latent array, but each latent is jointly modeled by all attention heads, leading to fully mixed representations that lack independent semantics.

Building on these observations, we propose **RATS** (**R**egister **A**ttention **T**ransformer**s**), aiming to let self-supervised learning spontaneously surface part-level representations. Concretely, we introduce an $L \to N \to N \to L$ bottleneck in each transformer block that decomposes the `[CLS]` aggregation into N register tokens: patches aggregate into registers (*compress*), registers exchange information (*communicate*), and broadcast back to patches (*broadcast*). The N registers are partitioned exclusively across the H attention heads, so each head owns an independent register subset that communicates only within the head. Confining each head to its own subspace encourages different heads to specialize to different regions, so part-level structure can emerge (Figure 1). On five segmentation benchmarks, RATS demonstrates stronger part-level grouping than all existing baselines ($+9.8$ mIoU at a matched token budget). By providing register tokens as initial query tokens, RATS further yields a stronger initialization for dense prediction, outperforming ViT on ADE20K ($+1.11$ mIoU) and COCO (AP^m $+0.2$). Finally, the visual dictionary extracted from the trained registers reveals a level semantic organization: within-supercategory part consistency and cross-category taxonomic proximity.

2 Related Work

Self-Supervised Visual Representation Learning and Segmentation. Self-supervised learning has progressed from pretext tasks [26, 51, 71] through contrastive [33, 13, 10] and non-contrastive [28, 15, 6] methods, to self-distillation on Vision Transformers [11] and masked image modeling [32, 5, 68, 73, 66]. The DINO family [11, 52, 56] has demonstrated that self-supervised training can yield patch tokens with strong part-level features alongside a global CLS token, as revealed by PCA visualization of the patch tokens. Several works [2, 49, 65, 36, 17, 30, 75, 3, 61] are related to part/object segmentation and discovery, but these methods fundamentally still extract features from

patch tokens. To the best of our knowledge, no existing method provides an architectural mechanism to explicitly extract part-level features through register tokens. RATS addresses this by introducing a register token bottleneck that extracts part-level information from patch tokens into register tokens.

Register Tokens in Visual Representation. Special-purpose learnable tokens have become a standard design pattern in transformers: the CLS token [22, 21] aggregates global semantics for classification, memory tokens [8] provide auxiliary storage for long-range context, and task-specific query tokens in DETR [9] attend to encoder features to produce object detections. In vision, Darcet *et al.* [19] show that introducing register tokens helps ViTs absorb and discard outlier information, stabilizing the representations of the remaining patch tokens. Building on this, Mamba-R [63] shows that Mamba-based vision models also suffer from the same artifacts and benefit from the same register token fix [70]. Latest models such as Dino v2/v3 [52, 56] and ViT-5 [62] have introduced registers as a default component. Concurrent work VECA [58] also places learned tokens at the attention bottleneck, but as an efficiency mechanism for linear-time ViTs rather than for part-level specialization. RATS repurposes registers entirely: by placing them at the bottleneck of attention and training with DINO self-distillation, each register is forced to specialize to a distinct visual part, turning a cleanup mechanism into a part-level representation primitive.

Learned Latent Queries for Visual Token Aggregation. Set Transformer [43] and Perceiver/IO [38, 37] established the $L \to N \to L$ cross-attention bottleneck pattern, but optimize supervised task objectives with no visual grouping semantics; Perceiver IO further entangles all latents through stacked $N \times N$ self-attention, preventing any latent from becoming an independent semantic unit. Slot Attention [45] instead uses learnable slot vectors that compete to represent scene elements via pixel reconstruction [7, 27, 23]; DINOSAUR [55] and AdaSlot [24] scale it to real images but remain at object level — reconstruction yields object-level alpha masks, and hard slot competition tears apart semantically coherent regions. Superpixel [48, 39] groups tokens by low-level visual cues but strict spatial locality prevents capturing parts requiring global context; GroupViT [69] achieves semantic grouping but requires image-text pairs. In all cases, no prior work has demonstrated the ability to spontaneously surface semantic part-level representations through self-supervised learning alone. RATS shows that the bottleneck paired with DINO self-distillation suffices to induce structured, transferable part-level primitives.

3 Method

RATS makes a single architectural change to a Vision Transformer: every block's self-attention is augmented with a bottleneck attention that decomposes global aggregation into N learnable register tokens, partitioned exclusively across the H attention heads. We first review the preliminaries (§3.1), then describe the bottleneck block, its training, and the resulting register visualization (§3.2).

3.1 Preliminaries

Vision Transformer. A Vision Transformer [22] splits an input image into L patches and adds a global classification `[CLS]` token, yielding a sequence $X \in \mathbb{R}^{(L+1)\times D}$ that passes through a stack of transformer blocks. With H heads and per-head dimension $d = D/H$, the multi-head self-attention (MHSA) [60] of a block is defined by

$$Q^{(h)} = \hat{X} W_Q^{(h)}, \;\; K^{(h)} = \hat{X} W_K^{(h)}, \;\; V^{(h)} = \hat{X} W_V^{(h)}, \;\; \mathrm{Attn}^{(h)} = \mathrm{softmax}\Big(\tfrac{Q^{(h)} K^{(h)\top}}{\sqrt{d}}\Big) V^{(h)}, \quad (1)$$

where $\hat{X} = \mathrm{LN}(X)$ is layer normalization [4]. The per-head outputs are concatenated and projected by a learned matrix $W_O \in \mathbb{R}^{D\times D}$, giving $\mathrm{MHSA}(\hat{X}) = \mathrm{Concat}(\mathrm{Attn}^{(1)}, \ldots, \mathrm{Attn}^{(H)})\, W_O$. Both sub-layers employ pre-normalization and residual connections: $X \leftarrow X + \mathrm{MHSA}(\mathrm{LN}(X))$ and $X \leftarrow X + \mathrm{MLP}(\mathrm{LN}(X))$.

DINO self-distillation. DINO [11] trains a student ViT f_{θ_s} to match the output of a teacher ViT f_{θ_t} on different augmented crops (views) of the same image: the teacher sees only the *global* crops $\mathcal{V}^g$, while the student sees both *global* and *local* crops $\mathcal{V} = \mathcal{V}^g \cup \mathcal{V}^l$. Both networks share the same architecture — a ViT backbone followed by an MLP projection head that maps the global `[CLS]` vector to a softmax distribution over K learnable prototypes. The student is updated by gradient

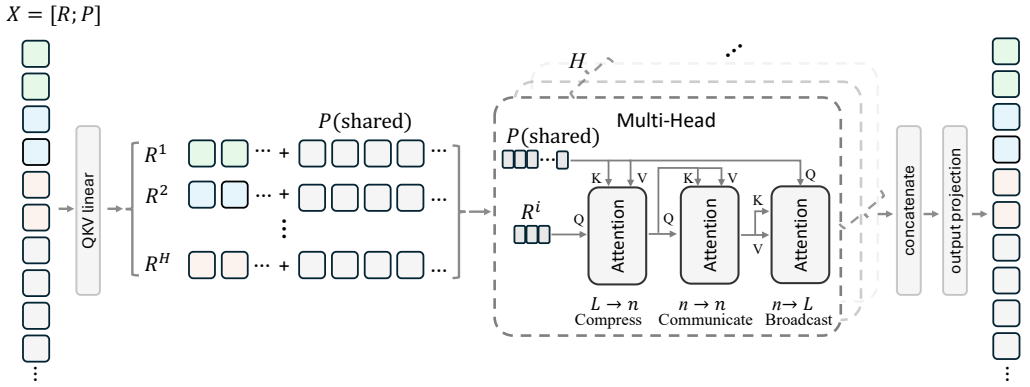


Figure 2: ***Register attention block.*** The shared QKV linear projects the input $X{=}[R; P]$ once. The register projections are partitioned into H disjoint per-head slices $R^1, \ldots, R^H$, while the patch projections P are shared across all heads. Each head runs the three-step attention — *compress*, *communicate*, *broadcast* — between its own register slice and the shared patches. Per-head outputs are concatenated and projected by the output projection W_O

descent, while the teacher's weights are an exponential moving average (EMA) of the student's, $\theta_t \leftarrow \lambda\,\theta_t + (1-\lambda)\,\theta_s$, where λ is the EMA decay rate. Writing $p_t(x)$ for the teacher's distribution on view x after centering and sharpening at a low temperature τ_t, and $p_s(x')$ for the student's distribution at a higher temperature τ_s, the DINO objective is a cross-entropy summed over all teacher-global to student-view pairs:

$$\mathcal{L}_{\text{DINO}} \;=\; \sum_{x \,\in\, \mathcal{V}^g} \; \sum_{\substack{x' \in \mathcal{V} \\ x' \neq x}} - \, p_t(x)^\top \log p_s(x'). \tag{2}$$

3.2 RATS: Register Attention Transformers

RATS introduces a bottleneck sub-block $\text{R-Attn}(\cdot)$ that decomposes the `[CLS]` token into N learnable *register tokens*. Block input is $X = [R;\, P] \in \mathbb{R}^{(N+L)\times D}$, where $R \in \mathbb{R}^{N\times D}$ are register tokens and $P \in \mathbb{R}^{L\times D}$ are patch tokens. Rather than aggregating all patches into a single `[CLS]` token, $\text{R-Attn}(\cdot)$ routes patch information through N register tokens via an $L{\to}N{\to}N{\to}L$ bottleneck, each operating in a low-rank subspace (rank $\leq n = N/H$ per head). This *compression* encourages each register to specialize to a proto-semantic, part-level region.

Single-head construction. Let n denote the per-head register count and d the per-head dimension ($n = N$ and $d = D$ when $H = 1$). Applying projection matrices $W_Q, W_K, W_V \in \mathbb{R}^{D\times d}$ to the LayerNormed input $\hat{X} = \text{LN}(X)$ yields register projections $Q_R, K_R, V_R \in \mathbb{R}^{n\times d}$ and patch projections $Q_P, K_P, V_P \in \mathbb{R}^{L\times d}$. The attention computation proceeds in three steps:

$$\textit{compress:}\;\; \hat{R} \;=\; \text{softmax}\!\left(\frac{Q_R K_P^\top}{\sqrt{d}}\right) V_P \;\in\; \mathbb{R}^{n\times d}, \tag{3}$$

$$\textit{communicate:}\;\; \bar{R} \;=\; \text{softmax}\!\left(\frac{\hat{R}\,\hat{R}^\top}{\sqrt{d}}\right) \hat{R} \;\in\; \mathbb{R}^{n\times d}, \tag{4}$$

$$\textit{broadcast:}\;\; O_P \;=\; \text{softmax}\!\left(\frac{Q_P\,\bar{R}^\top}{\sqrt{d}}\right) \bar{R} \;\in\; \mathbb{R}^{L\times d}. \tag{5}$$

Compress aggregates patch information into registers, *communicate* exchanges context among registers, and *broadcast* writes the updated registers back to every patch. Figure 3 visualizes this three-stage pipeline. Intuitively, *communicate* is self-attention among the registers: since each register already summarizes a coherent region, attending between them lets co-occurring parts exchange context and bind into larger structures—without it, patches would interact only indirectly through the compress–broadcast path, too weakly to capture patch–patch dependencies. Following §3.1, R-Attn replaces only the self-attention sub-layer; it keeps the same pre-norm residual connections and MLP, and the block outputs the updated tokens $[R;\, P]$.

Multi-head extension. In the Figure 2, with H heads, per-head dimension $d = D/H$, and per-head register count $n = N/H$, the N registers are partitioned exclusively across heads. Each head $h = 1, \ldots, H$ applies Eqs. (3)–(5) independently, using its own register slice $R^h = R[hn : (h{+}1)n]$ and the d-dimensional projection slice of the shared W_{QKV}. The outputs are assembled and projected:

$$\text{R-Attn}(\hat{X}) = \Big[\, O_R \;\Big|\; O_P \,\Big]\, W_O, \tag{6}$$

where $O_P = [O_P^1, \ldots, O_P^H] \in \mathbb{R}^{L \times D}$ concatenates heads along the feature axis, and $O_R \in \mathbb{R}^{N \times D}$ reassembles the per-head register outputs $R^1, \ldots, R^H$ into the original token and feature layout.

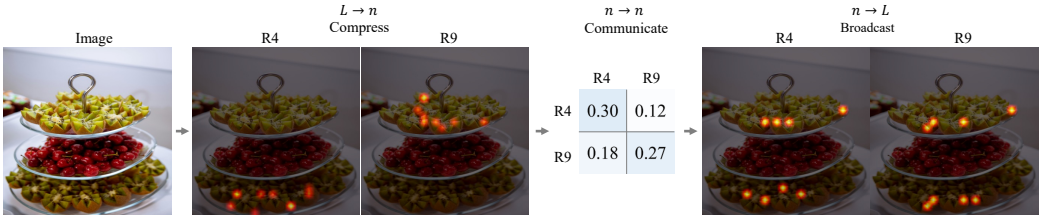


Figure 3: ***Visualization of the three-stage register attention block.*** *Compress*: registers R_4 and R_9 attend to distinct image regions (lower vs. upper tier). *Communicate*: self-attention among registers lets co-occurring parts exchange context and bind into larger structures. *Broadcast*: each register writes back to its own footprint, closing the $P{\to}R{\to}R{\to}P$ bottleneck and regrouping patches into proto-semantic regions.

Training. RATS is trained with the DINO objective (Eq. 2) without modification. The input sequence $X^{(0)} = [R;\, \text{PatchEmbed}(\mathbf{I}) + E_{\text{pos}}]$ passes through $\mathcal{L}$ pre-norm blocks of R-Attn and MLP with residual connections, where positional encoding E_{pos} is applied only to the patch tokens. The mean of the final-layer register tokens $z = \frac{1}{N}\sum_{i=1}^{N} R_i$ is fed to the DINO projection head g_{θ_s} as the global representation. The teacher shares the same architecture and is updated via EMA; no auxiliary region or part-level objective is used.

Register quality visualization. The similarity map between a register token and the patch tokens serves as a direct measure of the register's spatial selectivity: a highly localized and semantically coherent activation pattern indicates that the register has specialized to a visual part, while a diffuse or unstructured pattern suggests the register has not learned meaningful grouping. Accordingly, for each register R_i we compute its cosine similarity with every patch token from the last block, yielding a score map $S_i \in \mathbb{R}^{h \times w}$. We rank all N registers by their maximum activation and retain the top-τ most salient ones. The score maps are reshaped to the spatial grid and lightly smoothed with a Gaussian filter to reduce fragmentation, then each patch is assigned to its highest-scoring register via argmax to produce a segmentation map, as formalized in Eq. 7 below. Figure 1 (bottom) illustrates this pipeline and the resulting part-level similarity maps.

$$Seg(u,v) \;=\; \arg\max_{i \,\in\, \mathcal{T}} \big(G_\sigma * S_i\big)(u,v), \qquad S_i(u,v) \;=\; \frac{\langle R_i,\, P_{u,v}\rangle}{\|R_i\|\,\|P_{u,v}\|}. \tag{7}$$

where $\mathcal{T}$ indexes the top-τ registers ranked by $\max_{u,v} S_i(u,v)$, and G_σ is a 2-D isotropic Gaussian kernel with standard deviation σ.

From Registers to a Part Dictionary Each register attends to a specific region rather than summarizing the whole image, so its activation is a localized *part feature*. Because parts recur across images, these features cluster into a compact *dictionary* of part entries shared across the dataset. A standard ViT cannot expose such a dictionary: its single global `[CLS]` token provides no per-region tokens to aggregate. We construct it and evaluate its consistency and cross-category transfer quantitatively in §4.4; Figure 4 shows example entries.



Figure 4: ***Qualitative visualisation of individual dictionary entries.*** Each column shows attention overlays of a dictionary entry on two test images. *Left*: a single entry fires on the same part across different instances within a super-category. *Right*: Semantically similar dictionary entries (e.g., wheel-like entries) generalize across different super-categories.

4 Experiments

4.1 Implementation details.

We pretrain two model scales, RATS-S/B, on ImageNet-1k [20] at 224×224 using the DINO recipe without modification. We evaluate on five benchmarks: COCO 2017 val [44], ADE20K [72], ImageNet val, ImageNet-S_{919} [25], and PartImageNet [31]. Since ImageNet lacks segmentation masks and COCO instance annotations cover only a small fraction of each image, we adopt SAM 2.1 [41, 53] generated masks as proxy ground truth for these two datasets, and use native annotations for the remaining three. Note that the granularity of what counts as a "part" differs across these datasets, ranging from scene regions (ADE20K) and object classes (ImageNet-S_{919}) to object parts (PartImageNet). We report many-to-one mIoU (M2O) and ARI as primary metrics. For downstream transfer, we fine-tune with Mask2Former on ADE20K semantic segmentation and COCO detection/instance segmentation, comparing against DINO ViT-S/16 under identical recipes. Full details are provided in Appendix F.

4.2 Region Part Segmentation

We compare RATS against representative self-supervised ViT backbones (DINOv1 [11], DINOv3 [56]), Mamba [29, 74, 64] Backbone pretrained with register [70], superpixel-/region-oriented methods (SPFormer [48], SPiT [1]), and a recent slot-based model (AdaSlot [24]). For register-less ViTs (official DINOv1/v3 checkpoints), we split the final-layer `[CLS]` token across its attention heads and use each head's slice as a register query. We report many-to-one mIoU (%) on five benchmarks and the average ARI. As shown in Table 1, RATS establishes a new state of the art on every dataset and improves the average mIoU by **+9.8** over the strongest baseline (AdaSlot) at a matched token budget, consistent with the design goal of the register bottleneck to produce semantically coherent, part-aware regions. Figure 5 corroborates this qualitatively: where the baselines either fragment the foreground or leak into the background, RATS produces clean, grounded regions.

4.3 Downstream Transfer

The preceding experiments demonstrate that RATS's register bottleneck discovers semantically coherent part-level representations. A natural question is whether the register tokens can serve as input queries for query-based decoders such as Mask2Former [16]. To test this, we add the $L\to N\to N\to L$ bottleneck to a standard ViT and use the resulting register tokens as queries for Mask2Former, fine-tuning on ADE20K semantic segmentation and COCO 2017 detection/instance segmentation. We compare against the same architecture initialized from a DINO ViT-S/16 checkpoint under identical recipes (full details in Appendix F). As shown in Table 2, RATS pretraining consistently outperforms the DINO baseline on both ADE20K and COCO. The parenthesized values report results when replacing only the backbone, while keeping the original independently learned queries

Method	Backbone	#Register	COCO†	ADE20K	IN†	IN-S_{919}	PartIN	Avg	ARI
Standard ViT									
DINOv1	ViT-S/16	6	8.53	15.43	11.21	48.28	25.58	21.81	8.7
DINOv3	ViT-S/16	6	8.17	13.69	11.15	50.25	27.67	22.19	11.1
DINOv1	ViT-B/16	12	8.55	15.87	11.05	46.94	25.82	21.65	7.9
DINOv3	ViT-B/16	12	10.49	18.46	14.13	56.43	32.57	26.41	12.3
Mamba with Register									
DINOv1	Mam.R-B/16	12	10.07	18.95	13.44	55.54	30.11	25.62	12.8
Superpixel									
SPFormer	ViT-S/16	6	8.37	14.65	12.10	51.92	27.21	22.85	11.5
SPiT	ViT-B/16	12	8.41	12.89	12.37	53.47	27.83	23.00	8.5
Slot									
AdaSlot	ViT-B/16	33	12.31	23.87	15.49	54.83	30.74	27.45	19.3
Ours									
DINOv1	RATS-S/16	6	9.91	17.02	13.97	58.84	33.93	26.73	20.4
DINOv1	RATS-B/16	12	12.99	21.45	17.31	67.89	41.06	32.14	22.0
DINOv1	RATS-B/16	33	16.89	27.52	21.32	73.62	46.94	37.26	21.1
DINOv1	RATS-S/16	96	**17.31**	**26.98**	**21.87**	**73.18**	**46.43**	**37.15**	**21.7**
DINOv1	RATS-B/16	192	**19.32**	**30.77**	**23.47**	**75.39**	**48.77**	**39.54**	**19.6**

Table 1: ***part segmentation.*** M2O mIoU (%) on five benchmarks and mean ARI. The granularity of ground-truth "parts" differs across benchmarks, ranging from scene regions (ADE20K) and object classes (IN-S_{919}) to object parts (PartIN). RATS outperforms self-supervised, superpixel, and slot-based baselines at matched token budgets, and surpasses AdaSlot with fewer tokens. †Scored against SAM 2.1 proxy masks; the other three benchmarks use human annotations.

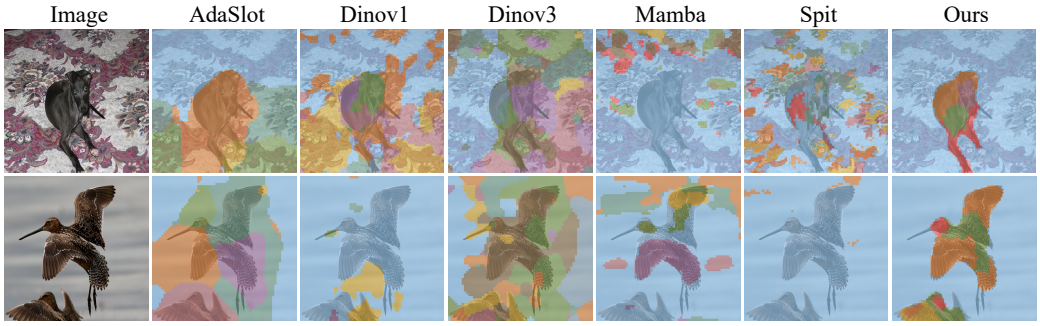


Figure 5: ***Qualitative comparison of part segmentation.*** RATS yields cleaner foreground–background separation and more part-coherent regions than the baselines. Extended comparison in Appendix B.

unchanged, thereby isolating the backbone contribution. This reveals that RATS's gains stem from two complementary sources: the bottleneck training produces stronger backbone features, while the register tokens provide part-level semantic representations that serve as superior query inputs.

		ADE20K (sem. seg.)			COCO 2017 (det. & inst. seg.)	
Pretraining	#queries	aAcc (%)	**mIoU (%)**	mAcc (%)	AP^b (%)	**AP^m (%)**
DINO	100	81.37	45.41	58.23	41.1	37.9
RATS	96 (100 init queries)	**81.61** (81.47)	**46.52** (45.83)	**59.37** (60.44)	**41.2**	**38.1**

Table 2: ***Downstream transfer to ADE20K and COCO 2017.*** Fine-tuning Mask2Former [16] with ViT-S/16 backbone, initialized from DINO or RATS pretraining (both 100 epochs on ImageNet-1k). Parenthesized values denote results when replacing only the backbone, while keeping the standard learned queries unchanged.

4.4 Dictionary Analysis

RATS's registers produce strong part-level features; we ask whether these can be aggregated into a dictionary of consistent, shared parts per category, and evaluate this on PartImageNet.

Codebook construction. A register that activates on a spatial region attends to that region in its attention map, yielding a localised part feature. We compare four codebook constructions in Fig. 6 (left). *Post-hoc K-means* serves as an upper bound: it extracts register features within ground-truth part masks and clusters them offline. *Learned codebook* randomly initialises a set of entries and

trains them on ImageNet by minimising the distance between each register and its nearest entry; ground-truth masks are used only afterwards to label each entry with its dominant part. The learned codebook trails K-means on retrieval by a small margin while delivering higher per-entry purity, confirming that RATS's register features are already well-structured for part discovery. Enlarging the vocabulary (*Learned codebook (L)*) further raises purity but wastes 90% of entries, exposing a purity–utilisation trade-off. We assess the dictionary along two axes: image-level retrieval (mAP, R@1) under a bag-of-entries representation, and per-entry weighted purity with utilisation (valPw, util%); see Appendix D for definitions.

Method	mAP ↑	R@1 ↑	valPw ↑	util% ↑
Random baseline	0.16	0.16	0.150	–
Post-hoc K-means	**0.56**	**0.93**	0.306	**100**
Learned codebook	0.48	0.83	0.317	62
Learned codebook (L)	0.45	0.82	**0.355**	10



Figure 6: ***Codebook construction and compositional generalisation.*** *Left*: retrieval and purity metrics under different codebook constructions. *Right*: a *Pegasus* (unseen composite of horse and bird wings) is decomposed by three independently learned entries, supporting zero-shot decomposition of novel compositions.

Qualitative verification. Figure 4 shows that individual entries fire consistently on the same part within a super-category (left) and that functional concepts such as "wheel" transfer across super-categories (right). Fig. 6 (right) further shows that a *Pegasus*—an unseen composite of horse and bird wings—is decomposed by independently learned entries, supporting zero-shot compositional generalisation. Additional analysis is in Appendix D.

4.5 Ablation Studies

Design Choices for Register-Based Token Aggregation. We ablate three axes: (i) per-head partition vs shared registers (all heads attend to the full register set); (ii) parameter-free vs learned communicate projections; (iii) RATS vs a Perceiver-IO style structure [37] that compresses patches into 80 latents, processes them with 10 self-attention layers, then decodes via 96 output queries. Table 3 reports cost (NVIDIA A5000 GPU, 512×512, batch=16) and quality averaged over five datasets at top-τ=6. RATS achieves lower latency and memory than ViT-S/16 at identical parameter count with improved segmentation quality, Pareto-dominating all design alternatives.

Model	Params	Mem	Latency	rel. cost	M2O ↑	O2O ↑	ARI ↑
ViT-S/16 (DINO)	21.67 M	1.08 GB	172.8 ms	2.12×	21.8	19.3	8.7
RATS (Baseline)	21.70 M	0.49 GB	81.7 ms	1.00×	**26.6**	**23.1**	**19.2**
RATS + learned comm	22.59 M	0.49 GB	82.6 ms	1.01×	25.5	21.9	17.1
RATS, shared	21.70 M	0.52 GB	91.7 ms	1.12×	25.0	21.2	15.1
Perceiver-IO style	21.73 M	0.31 GB	12.4 ms	0.15×	20.2	18.6	10.0

Table 3: ***Ablation on attention design choices.*** Each row varies one design axis from RATS (Baseline). Perceiver-IO minimises cost but sacrifices spatial detail.

Number of per-head registers n and over-segmentation. The register bottleneck compresses patch information into $N{=}n{\times}H$ region tokens, directly controlling grouping granularity. We ablate $n \in \{8, 16, 24, 32\}$ on RATS-S (H=6) at top-τ=40 and report both many-to-one (M2O) and one-to-one (O2O) mIoU. M2O reflects clustering quality, while a large M2O–O2O gap indicates that one semantic class is fragmented across multiple registers (over-segmentation). As shown in Table 4, n=16 achieves the best M2O, while smaller n only leads to a slight drop. Larger n tends to increase over-segmentation, whereas smaller n produces coarser but more complete regions.

Top-τ region ablation. At inference, we rank registers by activation magnitude and select the top-τ to form the segmentation map. Table 5 reveals a coverage-purity trade-off: increasing τ from 10 to 70 improves mIoU across all five datasets, with gains of up to 9.17 percentage points on IN-S919. While ARI initially improves on COCO, ADE20K, and ImageNet, it consistently declines beyond $\tau = 30$ across all datasets. Both metrics largely saturate beyond $\tau \approx 50$, suggesting a natural ranking among visual primitives learned by the bottleneck.

n	mIoU M2O (%) ↑						mIoU O2O (%) ↑					
	COCO	ADE20K	ImageNet	IN-S919	PartIN	Avg	COCO	ADE20K	ImageNet	IN-S919	PartIN	Avg
8	15.67	23.97	19.96	71.47	45.71	35.36	14.70	21.03	17.45	**43.10**	**32.89**	**25.83**
16	16.77	**26.59**	**21.37**	**72.72**	45.67	**36.62**	**15.81**	22.76	**18.34**	40.28	30.82	25.60
24	**16.89**	26.46	21.19	72.30	46.14	36.60	15.70	**22.85**	**18.34**	39.65	30.87	25.48
32	16.13	25.19	20.82	71.81	**46.22**	36.03	15.65	22.25	18.23	39.79	31.53	25.49

Table 4: ***Effect of per-head register count n and over-segmentation.*** M2O vs. O2O mIoU (%) on RATS-S. Higher M2O reflects better clustering; a large M2O–O2O gap indicates over-segmentation. n=16 achieves the best M2O, while n=8 trades absolute M2O for higher O2O on part-centric classes.

Top_τ	mIoU (M2O, %) ↑					ARI ↑				
	COCO	ADE20K	ImageNet	IN-S919	PartIN	COCO	ADE20K	ImageNet	IN-S919	PartIN
10	11.76	20.07	15.99	64.00	37.72	20.30	23.55	23.22	**19.74**	**24.00**
30	15.99	25.58	20.66	71.88	44.67	**23.01**	**24.77**	**24.74**	17.78	21.22
50	17.07	26.87	21.70	73.01	46.18	22.88	24.44	24.39	17.05	20.21
70	**17.28**	**26.97**	**21.85**	**73.17**	**46.41**	22.86	24.39	24.32	16.89	20.12

Table 5: ***Effect of top-τ selection.*** Larger τ improves spatial coverage (higher mIoU) but dilutes clustering purity (lower ARI), revealing a coverage-purity trade-off.

DINO loss placement. We investigate whether each attention head benefits from an independent DINO loss to learn distinct patterns, or whether a single loss over all registers suffices. As shown in Table 6, applying a separate DINO loss per head degrades mIoU, indicating that a shared loss over all registers is more effective.

One-to-one matching evaluation. We adopt a one-to-one (O2O) matching protocol, where each predicted part can match at most one ground-truth part, and unmatched predictions contribute nothing to the score. As shown in Table 7, RATS outperforms all baselines at both token budgets. These results suggest that RATS learns more coherent and complete object parts rather than fragmenting them into multiple predictions.

Greedy region merging evaluation. We greedily merge predicted regions for each ground-truth class to maximize its IoU (greedy mIoU). On ImageNet-S_{919}, RATS-S and RATS-B combine an average of 7.28 and 9.17 predicted regions to cover each ground-truth class, achieving greedy mIoU scores of 75.00% and 76.95%, respectively (Table 8). These results suggest that the fine-grained parts learned by RATS can be composed to form more complete object representations.

Effect of inference resolution. DINO-based models naturally support variable-resolution inputs at inference. Using RATS-S as a representative model, we sweep resolution from 224 to 1024 and report the mean across checkpoints at epoch 140–200, with the shaded band showing the min–max range (Figure 7). mIoU increases steadily up to 640, as finer patch grids provide more spatial detail for the register bottleneck. Beyond 640, gains saturate on most datasets while ARI begins to decline on ADE20K and PartIN, indicating that excessive patch counts dilute attention coherence.

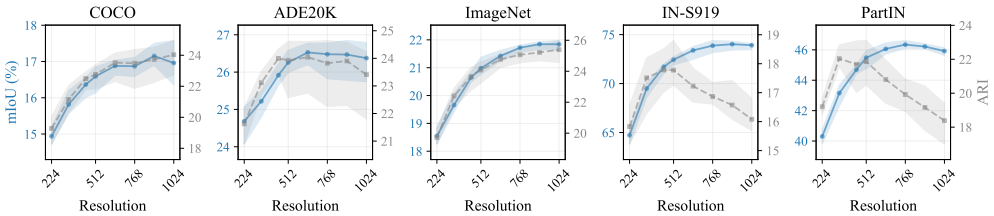


Figure 7: ***Effect of inference resolution (RATS-S).*** mIoU (M2O, %, blue, left axis) and ARI (gray, right axis) across resolutions 224–1024. Lines show the mean over epoch 140–200; shaded bands show the min–max range. mIoU improves up to 640 then saturates; ARI peaks at intermediate resolutions, revealing a resolution–coherence trade-off.

Training dynamics. We track mIoU of RATS-S across training epochs to understand when region structure emerges (Figure 8). Performance rises rapidly in early training, reaching ∼85% of peak by

Loss Scope	mIoU (M2O, %) ↑ COCO	ADE20K	ImageNet	IN-S919	PartIN
Per-head	15.38	21.82	18.47	67.20	39.76
All-head	**16.77**	**26.59**	**21.37**	**72.72**	**45.67**

Table 6: ***DINO loss placement.*** mIoU (M2O). *Per-head*: each attention head's register tokens receive a separate DINO distillation loss. *All-head*: a single shared DINO loss over all registers.

Method	Backbone	#Tokens	O2O (%) ↑
DINOv1	ViT-S/16	6	19.32
DINOv3	ViT-S/16	6	19.38
SPFormer	ViT-S/16	6	19.52
AdaSlot	ViT-B/16	33	21.39
RATS	RATS-S/16	6	23.10
RATS	RATS-B/16	33	**25.33**

Table 7: ***Average O2O mIoU over five benchmarks:*** RATS achieves the best results at both token budgets.

Model	Top-τ	Greedy mIoU (%) ↑	Pred. regions per GT class
RATS-S/16	6	63.89	2.41
RATS-S/16	96	75.00	7.28
RATS-B/16	12	70.64	4.00
RATS-B/16	192	**76.95**	9.17

Table 8: ***Greedy mIoU on ImageNet-S$_{919}$.*** Predicted regions are greedily merged to best cover each ground-truth class, showing how many parts RATS uses to compose an object.

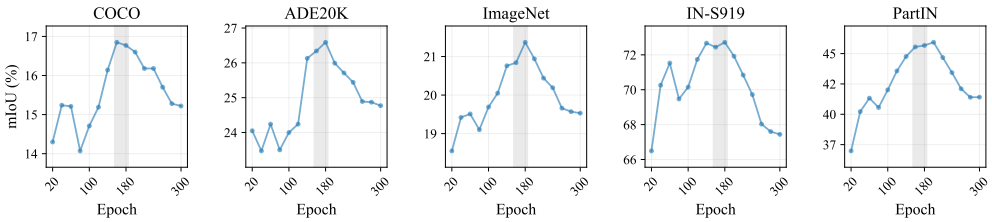


Figure 8: ***Training dynamics (RATS-S).*** mIoU (M2O) across epochs 20–300 on five benchmarks. The gray band marks the optimal window (epoch 160–180).

epoch 20. All five datasets peak around epoch 160–180, then gradually decline through epoch 300. This window coincides with the characteristic hump of the DINO loss, which a standard ViT-S also exhibits, and the decline is specific to the small RATS-S: RATS-B shows no such drop and reaches its best result at the final epoch (Appendix C.3).

5 Conclusion

We present RATS, a Vision Transformer architecture built on a register-token bottleneck that enables self-supervised models to spontaneously discover part-level representations without any part annotation or auxiliary loss. Experiments show that this simple architectural design surpasses existing methods on part segmentation and provides stronger features and query initialization for downstream dense prediction. The visual dictionary extracted from the trained registers further reveals structured semantic organization, including within-super-category part consistency and cross-category taxonomic proximity. Our results demonstrate that architectural inductive bias alone can serve as an effective prior for part discovery, providing a promising foundation for fine-grained recognition, interpretable visual analysis, and broader compositional visual understanding.

Acknowledgments and Disclosure of Funding

This work was supported by the Office of Naval Research (ONR) under grant N000142512132. The authors declare no competing interests.

References

[1] Marius Aasan, Odd Kolbjørnsen, Anne Schistad Solberg, and Adín Ramirez Rivera. A spitting image: Modular superpixel tokenization in vision transformers. In *European Conference on Computer Vision*, pages 124–142. Springer, 2024.

[2] 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.

[3] 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.

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[5] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. 2021.

[6] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. 2021.

[7] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

[8] Mikhail S Burtsev, Yuri Kuratov, Anton Peganov, and Grigory V Sapunov. Memory transformer. *arXiv preprint arXiv:2006.11527*, 2020.

[9] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[10] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. volume 33, pages 9912–9924, 2020.

[11] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[12] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark, 2019.

[13] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020.

[14] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1971–1978, 2014.

[15] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

[16] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022.

[17] Subhabrata Choudhury, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Unsupervised part discovery from contrastive reconstruction. volume 34, pages 28104–28118, 2021.

[18] MMSegmentation Contributors. Mmsegmentation: Openmmlab semantic segmentation toolbox and benchmark, 2020.

[19] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. 2023.

[20] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[22] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[23] Martin Engelcke, Adam R Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. 2019.

[24] Ke Fan, Zechen Bai, Tianjun Xiao, Tong He, Max Horn, Yanwei Fu, Francesco Locatello, and Zheng Zhang. Adaptive slot attention: Object discovery with dynamic slot number. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23062–23071, 2024.

[25] Shanghua Gao, Zhong-Yu Li, Ming-Hsuan Yang, Ming-Ming Cheng, Junwei Han, and Philip Torr. Large-scale unsupervised semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 45(6):7457–7476, 2022.

[26] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. 2018.

[27] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International conference on machine learning*, pages 2424–2433. PMLR, 2019.

[28] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. volume 33, pages 21271–21284, 2020.

[29] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[30] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T Freeman. Unsupervised semantic segmentation by distilling feature correspondences. 2022.

[31] 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, 2022.

[32] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[33] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[34] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016.

[35] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[36] Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz. Scops: Self-supervised co-part segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 869–878, 2019.

[37] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. 2021.

[38] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021.

[39] Varun Jampani, Deqing Sun, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Superpixel sampling networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 352–368, 2018.

[40] Nicholas Jiang, Amil Dravid, Alexei Efros, and Yossi Gandelsman. Vision transformers don't need trained registers. *Advances in neural information processing systems*, 38:56557–56595, 2026.

[41] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[42] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[43] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pages 3744–3753. PMLR, 2019.

[44] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[45] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. volume 33, pages 11525–11538, 2020.

[46] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. 2016.

[47] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. 2017.

[48] Jieru Mei, Liang-Chieh Chen, Alan Yuille, and Cihang Xie. Spformer: Enhancing vision transformer with superpixel representation. *arXiv preprint arXiv:2401.02931*, 2024.

[49] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8364–8375, 2022.

[50] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. 2017.

[51] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

[52] 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.

[53] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[54] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[55] Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, et al. Bridging the gap to real-world object-centric learning. 2022.

[56] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

[57] Sivic and Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings ninth IEEE international conference on computer vision*, pages 1470–1477. IEEE, 2003.

[58] Alan Z Song, Yinjie Chen, Mu Nan, Rui Zhang, Jiahang Cao, Weijian Mai, Muquan Yu, Hossein Adeli, Deva Ramanan, Michael J Tarr, et al. Elastic attention cores for scalable vision transformers. *arXiv preprint arXiv:2605.12491*, 2026.

[59] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. volume 30, 2017.

[60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. volume 30, 2017.

[61] Feng Wang, Jieru Mei, and Alan Yuille. Sclip: Rethinking self-attention for dense vision-language inference. In *European conference on computer vision*, pages 315–332. Springer, 2024.

[62] Feng Wang, Sucheng Ren, Tiezheng Zhang, Predrag Neskovic, Anand Bhattad, Cihang Xie, and Alan Yuille. Vit-5: Vision transformers for the mid-2020s. *arXiv preprint arXiv:2602.08071*, 2026.

[63] Feng Wang, Jiahao Wang, Sucheng Ren, Guoyizhe Wei, Jieru Mei, Wei Shao, Yuyin Zhou, Alan Yuille, and Cihang Xie. Mamba-reg: Vision mamba also needs registers. pages 14944–14953, 2025.

[64] Feng Wang, Timing Yang, Yaodong Yu, Sucheng Ren, Guoyizhe Wei, Angtian Wang, Wei Shao, Yuyin Zhou, Alan Yuille, and Cihang Xie. Adventurer: Optimizing vision mamba architecture designs for efficiency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 30157–30166, June 2025.

[65] 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 transactions on pattern analysis and machine intelligence*, 45(12):15790–15801, 2023.

[66] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14668–14678, 2022.

[67] Meng Wei, Xiaoyu Yue, Wenwei Zhang, Shu Kong, Xihui Liu, and Jiangmiao Pang. Ov-parts: Towards open-vocabulary part segmentation. *Advances in Neural Information Processing Systems*, 36:70094–70114, 2023.

[68] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9653–9663, 2022.

[69] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18134–18144, 2022.

[70] Timing Yang, Guoyizhe Wei, Alan Yuille, and Feng Wang. Rnn as linear transformer: A closer investigation into representational potentials of visual mamba models. *arXiv preprint arXiv:2511.18380*, 2025.

[71] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

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

[73] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. 2021.

[74] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. 2024.

[75] Adrian Ziegler and Yuki M Asano. Self-supervised learning of object parts for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14502–14511, 2022.

Appendix

A Limitations

RATS has three main limitations. **(i) Low-rank attention.** To allow registers to spontaneously acquire part-segmentation ability, we adopt the $L{\to}N{\to}N{\to}L$ structure. Because the $L{\to}N$ stage compresses information through only $n{=}N/H$ registers per head, the resulting attention has rank at most n, which is lower than the rank of standard $L{\to}L$ self-attention. For tasks that require maintaining a high-rank representation, such as semantic segmentation and object detection, we therefore introduce an additional parallel $L{\to}L$ branch during fine-tuning to recover full-rank capacity (Appendix. F). **(ii) Mismatch with human-defined semantics.** The grouping discovered by RATS is obtained without any annotation, so its part boundaries are determined by visual co-occurrence rather than by human-defined taxonomies. Compared with supervised segmentation, which targets human-labelled categories, the resulting regions may not always align with the parts a human would name, and the segmentation should be interpreted as an emergent visual decomposition rather than a label-faithful one. **(iii) Evaluation scope.** M2O can be inflated by over-segmentation and should be read together with O2O and ARI; and our downstream evaluation centers on dense prediction, leaving tasks such as semantic correspondence and depth estimation to future work. Whether part-level registers can be obtained at test time without retraining, in the spirit of Jiang *et al.* [40], is an interesting future direction.

B Extended qualitative comparison

Figure 9 extends the qualitative comparison of Figure 5 (main paper) with nine additional images spanning animals (bird, cocker spaniel), vehicles (bicycle, mountain bike), indoor/outdoor scenes (cathedral, bedroom), and challenging compositions (text-covered helmet, sculpture, person with prop). The same column layout is used: input image followed by AdaSlot [24], DINOv1 [11], DINOv3 [56], Mamba, SPiT [1], and RATS. Across all nine images, RATS consistently delivers cleaner foreground–background separation and more part-coherent regions than the baselines.

C Additional experiments

C.1 Additional part-annotated benchmarks

We further evaluate on two benchmarks with human part annotations, Pascal-Part [14] and ADE20K-Part234 [67]. At a matched budget of 33 registers, RATS leads AdaSlot on both (Table 9).

		Pascal-Part			ADE20K-Part234		
Method	#Register	M2O	O2O	ARI	M2O	O2O	ARI
AdaSlot	33	15.65	12.12	7.24	25.70	14.32	2.21
RATS-B/16	33	**23.25**	**17.17**	**14.81**	**30.79**	**17.52**	**2.67**

Table 9: ***Additional part-annotated benchmarks.*** M2O / O2O mIoU (%) and ARI at a matched budget of 33 registers.

C.2 Downstream transfer at larger scale

Table 10 extends Table 2 to ViT-B/16 and reports mean $\pm$ std over multiple fine-tuning seeds. RATS-S and RATS-B are pretrained for 100 and 400 epochs, respectively, with matched DINO baselines. On ImageNet-1k classification RATS is on par with ViT (linear probe / k-NN top-1: RATS-S 74.1/69.6 vs. ViT-S 74.0/69.3; RATS-B 78.3/76.3 vs. ViT-B 78.2/76.1), with gains concentrated on dense tasks.

C.3 Training dynamics and the DINO loss

Fig. 8 shows that the mIoU of RATS-S peaks around epoch 160–180. This window coincides with the characteristic hump of the DINO self-distillation loss, which both RATS-S and a standard ViT-S exhibit (Table 11): the DINO objective aligns global, view-invariant targets, which over long training



Figure 9: ***Extended qualitative comparison on part region segmentation.*** Same column layout as Figure 5 in the main paper. RATS produces consistent foreground–background separation and part-coherent regions across diverse natural images.

pushes the global representation toward a single object-level summary and reduces the diversity retained by the register tokens. With only $N{=}96$ registers and $H{=}6$ heads, RATS-S has limited register diversity, so its mIoU eases back after the hump. RATS-B ($N{=}192$, $H{=}12$) shows the same loss hump but no peak-and-decline: its mIoU stays on a plateau and reaches its best value at the final epoch 400.

Pretraining	ADE20K (sem. seg.) aAcc	mIoU	mAcc	COCO 2017 AP^b	AP^m
ViT-S	81.28±0.09	45.26±0.17	58.15±0.36	41.02±0.10	37.81±0.08
RATS-S	**81.54**±0.07	**46.24**±0.30	**59.31**±0.36	**41.13**±0.08	**38.06**±0.06
ViT-B	81.98±0.21	46.31±0.23	59.64±0.19	43.75±0.20	40.61±0.16
RATS-B	**82.26**±0.17	**47.10**±0.32	**60.16**±0.31	**44.15**±0.18	**40.91**±0.11

Table 10: ***Downstream transfer over multiple seeds.*** Mask2Former fine-tuning on ADE20K and COCO 2017 (mean ± std).

Epoch	60	100	140	160–180	220	300
ViT-S	3.60	4.07	4.43	4.54→4.63	4.66	4.25
RATS-S	4.87	5.19	5.51	5.59→5.66	5.58	5.10

Table 11: ***DINO training loss of ViT-S and RATS-S.*** Both follow the same rise-and-decline trajectory; the mIoU peak of RATS-S (Fig. 8) falls in the loss-hump region.

D Dictionary Evaluation and Structure

This appendix expands on the dictionary analysis reported in Section 4.4 of the main paper. We first describe the image representation, metrics, and codebook variants used in Figure 6, then present the structural analysis behind the qualitative claims in Figure 4.

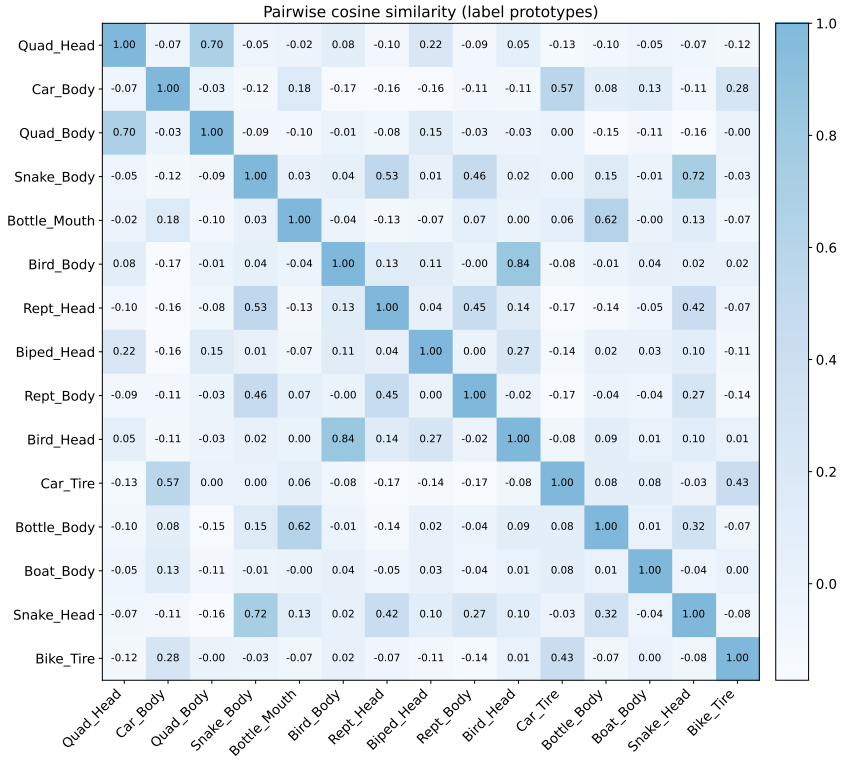


Figure 10: ***Pairwise cosine similarity between label prototypes.*** Each row/column is the averaged dictionary entry of one dominant part label. Three structures emerge—within-object coherence, taxonomic proximity, and functional analogy—analysed in Sec. D.

D.1 Qualitative analysis of the dictionary

Figure 11 shows register-attention overlays alongside ground-truth segmentations for six dictionary entries, each evaluated on three images from different classes within the same super-category. Each

entry consistently highlights the target part across varied instances, closely matching human-annotated boundaries, and the illustrated entries span both deformable biological parts (*Bird_Head*, *Fish_Head*, *Quad_Foot*) and rigid man-made parts (*Aero_Tail*, *Boat_Sail*, *Boat_Body*), confirming that a single learned vocabulary covers heterogeneous domains.

Beyond within-super-category consistency, the dictionary also captures cross-super-category structure. Figure 10 visualizes pairwise cosine similarity between label prototypes (averaged dictionary entries per dominant part label), revealing three semantic patterns. **Within-object coherence:** strongest off-diagonal similarities link parts of the same super-category (*Bird_Body–Bird_Head*: 0.84; *Snake_Body–Snake_Head*: 0.72), reflecting shared appearance context despite spatial disjointness. **Taxonomic proximity:** reptiles and snakes form a coherent block (cosines 0.42–0.53), and *Bird_Head–Biped_Head* reach 0.27, capturing cross-category biological similarity without part-label supervision. **Functional analogy:** rigid parts cluster by geometric function (*Car_Tire–Bike_Tire*: 0.43; *Bottle_Mouth–Bottle_Body*: 0.62), while unrelated parts remain near zero. Together, these patterns show the dictionary discovers multi-level semantic organisation purely from the register bottleneck.

Figure 12 qualitatively validates the latter two patterns. **Taxonomic transfer** (rows 1–2): *Snake_Head* and *Reptile_Head* entries transfer bidirectionally, localising heads across turtles, lizards, chameleons, and snakes despite large shape and texture differences. **Functional transfer** (rows 3–4): *Car_Tire* fires on bicycle wheels and *Bike_Tire* localises tyres on vans, buses, and race cars. These results confirm that high off-diagonal entries in Figure 10 reflect genuinely reusable part concepts rather than TF-IDF artefacts.

D.2 Codebook construction and metrics

Dictionary construction. We compare two ways of building the codebook of K entries: a *post-hoc K-means* dictionary built on PartImageNet, and a *learned* dictionary trained with the backbone on ImageNet. A common labelling procedure, described at the end of this section, attaches a part label to every entry after construction.

Post-hoc K-means dictionary. We extract register features $\{r_i\}$ from the last block of the trained RATS model on every image of the PartImageNet training split. For each register, we check whether its attention map peaks inside a ground-truth ROI mask; only registers whose dominant attention falls within a foreground part region are retained, filtering out registers that primarily attend to background. We then run K-means on the pooled foreground-filtered register features to obtain $K{=}512$ centroids $\{e_k\}_{k=1}^{K}$ that serve directly as the dictionary entries. PartImageNet contains 40 fine-grained part categories (11 super-categories with ~ 4 parts each) plus background; setting K roughly an order of magnitude above this label count gives each part several appearance-specific sub-prototypes (e.g. heads of different bird species, tyres of different vehicle models).

Learned dictionary. We initialise K entry vectors $\{e_k\}_{k=1}^{K}$ and optimise them jointly with the RATS backbone on **ImageNet** under the standard self-supervised objective. For each image, the registers $\{r_i\}_{i=1}^{N}$ from the last block are hard-assigned to their nearest entry, $k_i^* = \arg\min_k \|r_i - e_k\|_2$, and the entries are pulled toward their assigned registers via a commitment + codebook loss [59], with an entropy-based diversity term that prevents codebook collapse. Because training sees ImageNet (1000 classes, covering parts not annotated in PartImageNet) rather than PartImageNet's ~ 40 part labels, K is scaled to the pre-training distribution: $K{=}4096$ as our default, and $K{=}65{,}535$ for the large-vocabulary ablation.

Associating entries with semantic part labels. At this point the dictionary is just a set of K unnamed vectors. To turn it into a usable lexicon we need to attach a human-readable part name (e.g. *Bird_Head*) to each entry. The intuition is simple: *when a register lands in entry e_k, what part of the image does it tend to look at?* If almost every register that lands in e_k looks at bird heads, we call e_k a *Bird_Head* entry. We make this precise on the PartImageNet training split in three steps.

(i) Map each register to a dictionary entry. For every image, the N registers from the last block are hard-assigned to their nearest dictionary entry, $k_i^* = \arg\min_k \|r_i - e_k\|_2$. This step uses no part annotations; it is the same matching that the dictionary was built with.

(ii) Map each register to a ground-truth part. We now ask: which annotated part does this register attend to? Each register has a spatial attention map over the patch grid (from the L $\rightarrow$ N stage of the



Figure 11: ***Within-super-category consistency of individual dictionary entries.*** Each column corresponds to one dictionary entry; each of the three (ground-truth part segmentation, register-attention overlay) pairs shows a different PartImageNet class from the same super-category. The attention consistently aligns with the annotated part boundary across instances, and the six entries together span both biological (*Bird_Head*, *Fish_Head*, *Quad_Foot*) and rigid man-made (*Aero_Tail*, *Boat_Sail*, *Boat_Body*) super-categories, despite no part-level supervision being used during training.

RATS bottleneck), normalised to $[0, 1]$. The ground-truth segmentation provides a binary mask for every part in the same image. We overlay the attention map onto each part mask and compute the average attention value inside that mask. The part receiving the highest average attention is taken as the register's part label.

(iii) Per-entry voting. After the previous two steps, every register carries two labels: which dictionary entry it belongs to, and which ground-truth part it attends to. For each entry, we count the part labels of all registers assigned to it; the most frequent part becomes the entry's *dominant part*, and its proportion is the entry's *purity*. For example, if 100 registers land in an entry and 82 of them attend to *Bird_Head*, the entry is named *Bird_Head* with a purity of 0.82. Higher purity indicates that the entry more consistently corresponds to a single part. Entries receiving fewer than three registers are discarded, as too few samples cannot yield a reliable statistic.

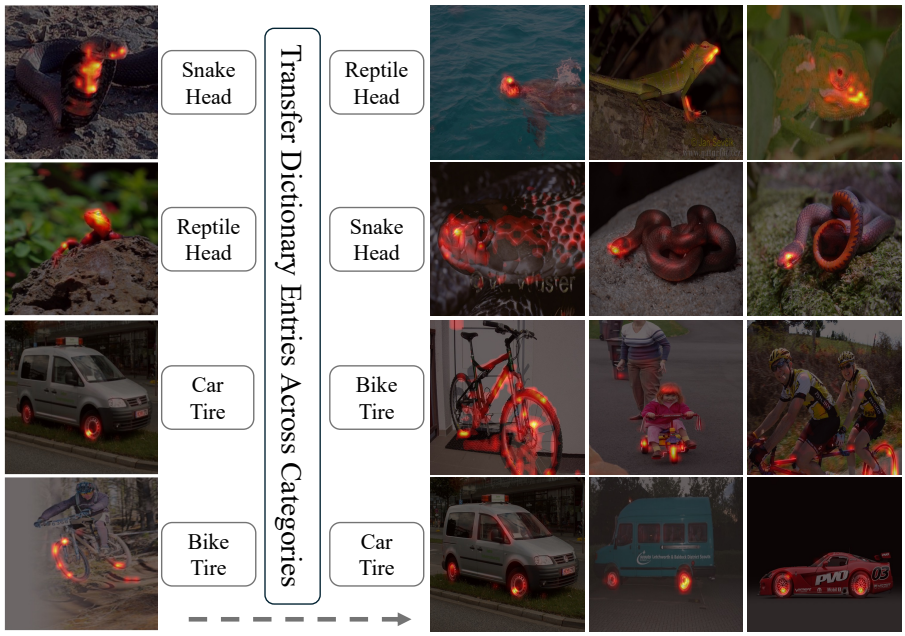


Figure 12: ***Cross super-category transfer of dictionary entries.*** The left column shows a source image whose registers activate a given entry (*Snake_Head*, *Reptile_Head*, *Car_Tire*, or *Bike_Tire*); the right three columns apply the *same* entry to three images drawn from a *different* super-category (labelled above the arrow). Rows 1–2 demonstrate taxonomic transfer between snake and reptile head entries (heatmap cosine ≈ 0.42–0.53); rows 3–4 demonstrate functional transfer of the wheel concept between car and bike tyre entries (heatmap cosine 0.43). The attention localises the analogous part in the target super-category.

Inference with the dictionary. Given a new image, we run a single forward pass through the RATS model to obtain each register's feature r_i and attention map A_i. Matching r_i to its nearest dictionary entry assigns the register the corresponding part label, and the attention map indicates where that part lies in the image. Since multiple registers may be assigned the same part (e.g. several registers all match to *Bird_Head*), we merge their attention maps to produce a heatmap that covers the full extent of that part in the image (see Fig. 4 and Fig. 6).

Bag-of-entries representation. Given the register-to-entry assignment described above, we can describe an image by which dictionary entries its registers fall into and how many land in each [57], forming a K-dimensional count vector $\mathrm{tf}_j \in \mathbb{R}^K$. Using raw counts directly would bias toward entries that appear frequently across all images (e.g. background). We therefore borrow the TF–IDF scheme [54] from information retrieval: the more images an entry appears in, the lower its weight ($\mathrm{idf}_k = \log \frac{|\mathcal{D}|}{|\{j : \mathrm{tf}_{j,k} > 0\}|}$, where $|\mathcal{D}|$ is the total number of images). After weighting and ℓ_2 normalisation, the final representation is $\mathbf{v}_j = (\mathrm{tf}_j \odot \mathrm{idf}) \,/\, \|\mathrm{tf}_j \odot \mathrm{idf}\|_2$. Image retrieval then ranks database images by cosine similarity against the query.

Metrics. *mAP* (mean Average Precision) measures whether the dictionary can help retrieve similar images. Concretely, we take each image, use its bag-of-entries vector to find the most similar images in the database, and check whether the retrieved images belong to the same object category as the query (e.g. both are birds, both are fish). Repeating this for all images and averaging gives mAP. *R@1* is a simplified variant that only checks whether the single most similar image shares the query's category; it saturates quickly on this benchmark and offers limited discriminative power.

valPw measures how specialised the dictionary entries are. Ideally, each entry should correspond to exactly one part (e.g. only bird heads). valPw is the weighted average of all entries' purities, where the weight of each entry is the number of registers assigned to it—entries matched by more registers are more important and thus count more in the average. Only entries matched by at least three registers are included.

util% measures how many of the K entries are actually used (i.e. matched by at least three registers). For example, if $K{=}4096$ but only 400 entries are used, util% is roughly 10%, indicating that most of the dictionary capacity is wasted.

E Per-head register attention maps

Figure 13 shows attention maps from three heads of the last block on two images containing dense object instances (a fruit grid in row 1 and a sliced-fruit row in row 2). For each head we display two registers and visualise their $L{\to}N$ attention overlaid on the image. Two qualitative observations stand out. **(i)** Within a single head, distinct registers focus on different but often related spatial regions — e.g., R_0 and R_{13} in Head 1 both fire on green-fruit clusters but at different positions. **(ii)** Across heads, the same image region is covered by different register combinations: Head 1 and Head 2 emphasise different inter-fruit boundaries, while Head 3 captures cross-row alignment. These patterns are consistent with the per-head register separation acting as an inductive bias toward diverse spatial decompositions of the input.

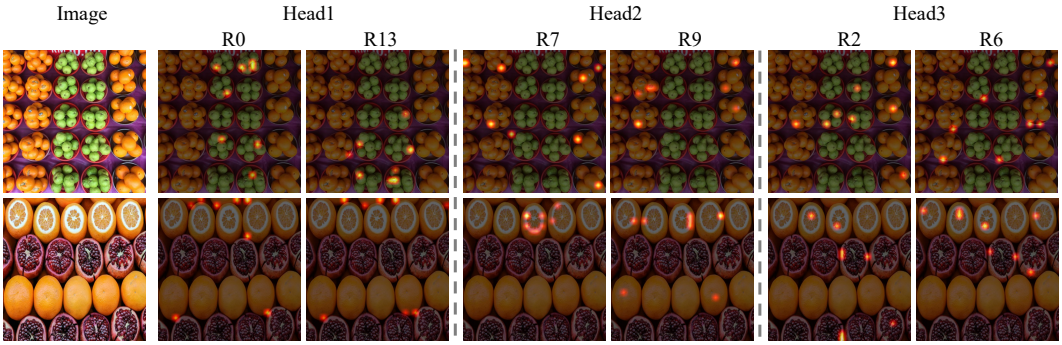


Figure 13: ***Per-head register attention maps.*** Six selected registers from three heads on two dense-instance images; analysed in Sec. E.



Figure 14: **Ground-truth annotations across the five evaluation benchmarks.** ImageNet and COCO use SAM 2.1 generated masks; ADE20K, ImageNet-S_{919}, and PartImageNet use their native annotations.

F Implementation details.

Architecture. We provide two model scales: RATS-S and RATS-B, based on ViT-S/16 ($D{=}384$, $H{=}6$) and ViT-B/16 ($D{=}768$, $H{=}12$), both with 12 layers. In each block, MHSA is replaced by the region bottleneck attention. Registers receive no positional encoding and are hard-partitioned into $n{=}N/H$ per head. By default, RATS-S uses $N{=}96$ ($n{=}16$) and RATS-B uses $N{=}192$ ($n{=}16$).

Pretraining. Both models are pretrained on ImageNet-1k at $224{\times}224$, following the DINO recipe: AdamW [47] optimizer, cosine weight decay from 0.04 to 0.4, linear warmup (10 epochs) followed by cosine learning rate decay [46], teacher EMA momentum $\lambda{=}0.996{\to}1.0$, output dimension $K{=}65{,}536$, two global crops (224^2) plus ten local crops (96^2), teacher temperature $\tau_t{=}0.07$, and bf16 mixed precision [50] throughout. RATS-B fully aligns with the official DINO ViT-B hyperparameters: 400 epochs, total batch size 1024, base learning rate $7.5{\times}10^{-4}$, minimum learning rate $2{\times}10^{-6}$, drop path [34] 0.1, gradient clipping 0.3, teacher temperature warmup for 50 epochs, and freeze last

layer for 3 epochs. RATS-S follows: 300 epochs, total batch size 768 (96 per GPU $\times$ 8 GPUs), base learning rate 2.5×10^{-4}, minimum learning rate 1×10^{-5}, drop path 0.05, gradient clipping disabled, teacher temperature warmup for 30 epochs, and freeze last layer for 1 epoch. The training objective is solely the DINO student–teacher cross-entropy.

Datasets and evaluation. Evaluating part discovery requires dense region annotations that cover the entire image. However, ImageNet lacks segmentation masks entirely, and COCO, while providing instance annotations, has too few masks with limited spatial coverage to adequately evaluate full-image region grouping. We therefore adopt SAM 2.1 as proxy ground truth for these two datasets: we run the official `sam2.1-hiera-large` automatic mask generator with a 128×128 point prompt grid to produce dense, category-agnostic region maps on each image. The remaining three datasets use their native annotations.

We evaluate on five benchmarks: COCO 2017 val (5,000 images, SAM masks), testing grouping in cluttered multi-object scenes; ADE20K validation (2,000 images), testing robustness under large semantic diversity; ImageNet val (50,000 images, SAM masks), testing part-level grouping on object-centric images; ImageNet-S_{919} val (10,767 images), biased toward foreground and part evaluation; and PartImageNet test (2,408 images), with human part annotations. Figure 14 shows representative examples with their ground-truth annotations from each benchmark.

Metrics. We report many-to-one mIoU (M2O) as the primary metric, which greedily assigns each predicted region to the ground-truth region with maximum overlap, allowing multiple predictions to map to the same ground truth. We additionally report ARI (Adjusted Rand Index) [35], which measures agreement between predicted and ground-truth groupings without relying on region-count matching. In ablation studies, we further report one-to-one mIoU (O2O, via Hungarian matching [42]): a large M2O–O2O gap indicates that the model fragments a single semantic region across multiple registers. All segmentation methods are evaluated at 512×512. Higher resolution provides finer patch grids for the register bottleneck while remaining computationally feasible for all baselines. We ablate resolution sensitivity in Figure 7.

Downstream transfer setting. We fine-tune the pretrained RATS and ViT-S/16 backbone with a Mask2Former [16] decoder on two benchmarks. For ADE20K semantic segmentation, we follow the standard MMSegmentation [18] recipe (crop 512×512, 160K iterations, batch size 8, 96 object queries). For COCO 2017 detection and instance segmentation, we follow the standard MMDetection [12] recipe (1024×1024 input, 368.75K iterations, batch size 8, 96 object queries, AdamW). The baseline uses the same Mask2Former architecture initialized from a DINO [11] ViT-S/16 checkpoint trained for the same 100 epochs, with all fine-tuning hyperparameters kept identical.

During fine-tuning, we restore a parallel $L\to L$ self-attention path alongside the register bottleneck, so that each block computes

$$X_{\text{out}} = \underbrace{(L\to N\to N\to L)(X)}_{\text{bottleneck}} + \underbrace{(L\to L)(X)}_{\text{full-rank}}. \tag{8}$$

The bottleneck path extracts part-level representations that serve as initial queries for the Mask2Former decoder. However, this path alone is low-rank: each head operates through only $n{=}N/H$ registers (e.g., $n{=}16$), producing an attention matrix of rank at most n, which is insufficient to maintain the full expressive capacity of a ViT whose representation rank scales with the patch sequence length L. The $L\to L$ path restores standard self-attention among all patch tokens, preserving full-rank capacity and ensuring that the fine-tuned model retains the dense representational power required for high-resolution segmentation and detection.

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 architectural design, segmentation gains, downstream improvements, and dictionary structure claimed in the abstract and Sec. 1 are each directly substantiated by the corresponding tables and figures in Sec. 4.

   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: Limitations are discussed in the appendix.

   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: The paper does not contain formal theoretical results; all claims are empirical.

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: The architecture, training recipe (DINO hyperparameters, optimizer, schedule, batch size, crops), evaluation protocols, and downstream fine-tuning details are fully specified in Sec. 3 and Appendix F.

   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: Code is publicly released at `https://github.com/yangtiming/RATS` with training and evaluation scripts; all datasets used (ImageNet, COCO, ADE20K, ImageNet-S_{919}, PartImageNet, Pascal-Part, ADE20K-Part234) are publicly available.

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: Sec. 4.1 summarises the setup and Appendix F provides the full pretraining recipe, dataset splits, evaluation resolution, metrics, and downstream Mask2Former finetuning recipes.

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: [Yes]

Justification: Downstream transfer results are reported as mean $\pm$ std over multiple finetuning seeds (Appendix C.2); Fig. 7 additionally shows min–max bands across checkpoints.

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: Tab. 3 reports per-image latency and memory on an NVIDIA A5000, and Appendix F specifies per-GPU batch sizes and training epochs for both RATS-S and RATS-B.

   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 have reviewed the NeurIPS Code of Ethics and the research uses only public benchmark datasets, with no human subjects or sensitive data involved.

   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: [N/A]

Justification: This is foundational research on self-supervised visual representation learning with no direct application or deployment, so we do not foresee specific societal impacts beyond those of generic visual backbone improvements.

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: The released models are visual recognition backbones trained on standard public datasets and pose no high risk of misuse.

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: All datasets (ImageNet, COCO, ADE20K, ImageNet-S_{919}, PartImageNet) and pretrained models (DINO, DINOv3, SAM 2.1, Mask2Former) are cited and used under their original terms of use.

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: The public code release (`https://github.com/yangtiming/RATS`) includes a README documenting environment setup, training, and evaluation commands for the proposed RATS models.

    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: The paper does not involve crowdsourcing or research with human subjects.

    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: The paper does not involve research with human subjects, so IRB approval is not applicable.

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: LLMs are not used as a component of the core methodology; any use is limited to writing and formatting assistance.

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.