WHAT-WHERE TRANSFORMER: CONCURRENT REPRESENTATION AND LOCALIZATION IN A SLOT-CENTRIC VISUAL BACKBONE

Anonymous authors
Paper under double-blind review

ABSTRACT

Image understanding tasks involve identifying *what* is present and *where* it appears. However, classification-oriented visual backbones primarily optimize semantic representations of *what*, while spatial information about *where* often remains implicit or entangled with semantic features. We hypothesize that explicit separation of these *what* and *where* representations can induce localization ability within backbones. To examine this, we introduce the What-Where Transformer (WWT), a ViT-style attentive backbone that incorporates what–where separation as an architectural inductive bias. Our method introduces two key designs: (1) it treats tokens as representations of *what* and attention maps as representations of *where*, and processes them in concurrent feed-forward modules via a multi-stream, slot-based architecture; (2) it reuses both the final-layer tokens and attention maps for downstream tasks, and directly exposes them to gradients derived from tasks, facilitating explicit learning of localization. Trained on ImageNet with single-label classification and an auxiliary reconstruction objective, WWT discovers multiple objects directly from its raw attention maps without token clustering or other postprocessing, at a modest cost to classification accuracy. Furthermore, WWT achieves superior performance compared to ViT-based methods on zero-shot object discovery and weakly supervised semantic segmentation, and it is transferable to various localization setups with minimal modifications.

1 INTRODUCTION

Transformers equipped with attention mechanisms (Vaswani et al., 2017) have proven effective in image recognition. However, although Vision Transformers (ViTs) serve as general-purpose visual encoders (Dosovitskiy et al.), localization is largely delegated to additional prediction architectures tailored to downstream tasks, as exemplified by DETR (Carion et al., 2020). This requirement increases the complexity of building unified visual systems that span multiple tasks. In contrast, embedding localization ability directly into the visual encoder could provide more flexible and transferable localization for downstream spatial reasoning.

Conceptually, the attention maps in a ViT encoder should capture object regions, and some studies have partially observed such behavior (Caron et al., 2021; Rawlekar et al., 2026). However, in practice, encoder attention often suffers from noisy activations, vague boundaries, and limited semantic interpretability (Wang et al., 2022; Psomas et al., 2023; Darcet et al., 2024). As a result, these attention maps are not easily transferable to downstream tasks. Many approaches decouple task decoders from encoder attention; the use of ViT encoder attention in downstream tasks is mostly limited to specific settings such as weakly supervised learning (Xu et al., 2022; Gupta et al., 2022).

We hypothesize that token-centric design of ViTs contributes to the limited transferability of encoder attention. ViT tokens, mostly derived from image-patch embeddings, serve as inputs, outputs, and internal states. Although attention maps temporarily capture spatial relationships between tokens, their role is limited to assisting token updates with spatial integration, and they are immediately absorbed back into the tokens within each block (Fig. 1a). In contrast, this work draws inspiration from the concept of what–where separation (Mishkin et al., 1983; Gregor & LeCun, 2010), in which the semantic content and spatial location of each visual entity are represented separately.

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

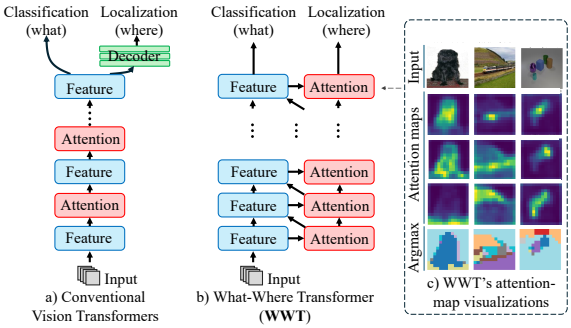


Figure 1: Concept-level illustrations of a) conventional Vision Transformers and b) proposed What-Where Transformer (WWT). WWT features multi-stream design separating what pathway and where pathway. This design facilitates emerging separation of representations and their locations; as a result, c) it learns multiple-object discovery in the attention maps even in single-label classification tasks.

To revive this classical concept in modern ViT, we introduce a design that pairs conventional token-like distributed representations with additional spatial representations, encouraging the model to associate them with "what" and "where," respectively. By incorporating what–where separation as an inductive bias (Goyal & Bengio, 2022) in the architecture and loss, the model can acquire explicit positional representations even under pretraining tasks that do not directly require location information. As a result, not only "what" but also "where" can be extracted from the output, which is expected to simplify task-specific decoders and improve the transferability of the learned models as backbones for downstream tasks.

As an instantiation of what-where separation in ViT-style attentive architectures, we propose novel What-Where Transformer (WWT), shown in Fig. 1b). The model maintains separate representations for object semantics ("what") and their spatial information ("where"), while enabling their concurrent interaction. Visual components in an image are represented as pairs consisting of a feature vector and a soft mask. The network is constructed by stacking multi-stream feedforward layers, which process these pairs individually, and inter-attention layers, which allow communication between streams. The soft mask not only aggregates spatial features like attention maps, but can also be directly interpreted as a region mask in the output. With relatively simple post-processing, it can be converted into boxes or dense labels, making it broadly applicable to localization tasks.

In the experiments, WWT was pretrained in a supervised manner on ImageNet-1k (Deng et al., 2009), and we confirmed the following: 1) In ImageNet classification, WWT produces substantially more interpretable output masks than ViT attention maps, at the cost of some classification accuracy compared to the corresponding DeiT models. 2) The output masks obtained from classification training can be directly applied to segmentation without retraining or additional modules. 3) It can also be extended to single- and multi-object discovery and supervised object detection with minimalistic modifications. Especially, multi-object discovery solely from attention remains challenging for existing ViTs, since class-token attention often fails to separate multiple object regions, even when the objects belong to different classes.

Our contributions are summarized as follows. 1) Design principle: We frame the long-standing idea of what-where separation and analyze its role as an inductive bias, and establish our core design, tokens-as-representations and attentions-as-localization. 2) Architecture: We develop WWT, a ViT-style attentive backbone that instantiates the principle. Through mutual attention, WWT replaces ViT's dense token-token self-attention with slot-mediated bipartite communication and explicitly propagates localization states across layers. 3) Emergent multi-object discovery: Under our architectural inductive bias, we observed that multi-object-decomposed attentions emerge from single-label image-level classification training, and evaluated zero-shot object discovery tasks. 4) Downstream versatility: WWT applies its backbone-level localization representations to multiple tasks with relatively lightweight adaptation. These results provide a proof of concept for what–where separation as a design principle for visual backbones with reusable, explicit spatial representations.

2 Related Work

Transformers in vision Vision Transformers (ViTs) have successfully adapted Transformer encoders for image classification and representation learning (Vaswani et al., 2017; Dosovitskiy et al.). Beyond scaling plain ViTs (Dehghani et al., 2023), many variants introduce multi-scale representations (Wang et al., 2021), local attention (Liu et al., 2021b), convolutional operations (Wu et al., 2021), or learnable queries (Yao et al., 2023). These architectures can be broadly understood through the MetaFormer framework, which interleaves spatial and channel mixing (Liu et al., 2022; Yu et al., 2023). While various spatial mixers have been explored, including self-attention, pooling, spatial MLPs (Liu et al., 2021a; Touvron et al., 2022), and clustering (Ma et al., 2023), most architectures

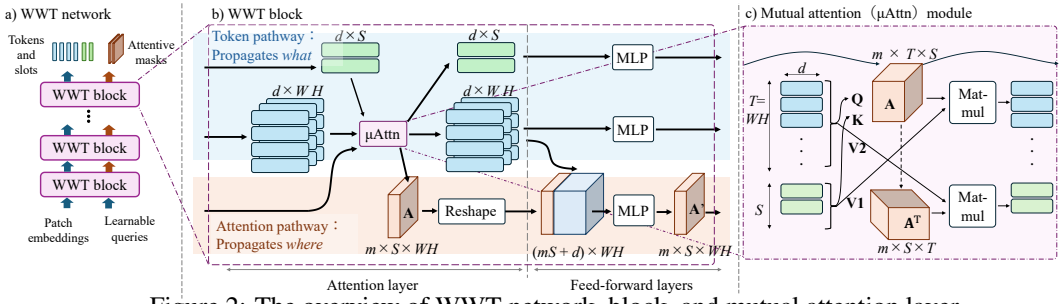


Figure 2: The overview of WWT network, block, and mutual attention layer.

remain token-centric. Perceptual grouping (Yu et al., 2002) can likewise serve as a spatial mixer through hierarchical or iterative token grouping (Ke et al., 2024; Deng et al., 2024; Brasó et al., 2025). These methods group or aggregate visual tokens, whereas WWT maintains separate but interacting representations of semantic content and spatial support throughout the backbone.

Transformers are also widely used as task decoders, as in DETR and MaskFormer (Carion et al., 2020; Cheng et al., 2021), with many subsequent variants (Zhu et al., 2021; Zhang et al., 2023; Cheng et al., 2022). Other approaches pair Transformer encoders with lightweight heads for dense prediction, reusing general-purpose representations while limiting task-specific computation (Ranftl et al., 2021; Li et al., 2022). This design is particularly attractive for large-scale pretrained vision and vision–language models (Minderer et al., 2022; Kirillov et al., 2023; Li et al., 2025). WWT brings learnable queries and cross-attention-like interactions, commonly associated with task decoders, into a localization-aware visual backbone.

Slot Attention Slot attention (SA) learns object-centric representations by assigning image features to latent slots, typically through an autoencoding objective that supports unsupervised object segmentation (Locatello et al., 2020). Subsequent work has explored Transformer-based slot updates or decoders (Singh et al., 2022; Wu et al., 2023), diffusion decoders (Jiang et al., 2023), classification objectives (Li et al., 2021; Wang et al., 2023; 2024), pretrained visual encoders (Caron et al., 2021; Seitzer et al., 2023), and slot extraction and fusion across multiple encoder layers (Bock et al., 2026). These methods retain a separate image encoder, even when slot learning accesses multiple encoder layers. In contrast, WWT embeds slots and their spatial supports directly into the backbone, where they evolve concurrently with patch features across layers.

What-where separation The idea of what-where separation has two independent origins: In neuroscience, the two-stream hypothesis (Mishkin et al., 1983; Milner & Goodale, 1995) proposed distinct pathways for object recognition ("what") and spatial processing ("where" or "how"). This inspired two-stream CNNs modeling what-where separation through foveated or retinal transformations (Choi et al., 2023; Ebrahimpour et al., 2019), but they relied on parallel CNN pathways rather than representational factorization. Independently, early representation learning raised concerns that CNNs discard positional information through pooling (Hinton et al., 2011). Subsequent models addressed this by encoding pose transformations (Sabour et al., 2017), or preserving reference positions during pooling (Zhao et al., 2016). Multi-object VAEs, precursors to object-centric learning, use attention maps to define responsibility regions for reconstruction (Burgess et al., 2019); however, their latent variables remain global image-level vectors. An orthogonal line of work has focused on improving positional embeddings and phase-based representations (Liu et al., 2025; Gopalakrishnan et al., 2025). WWT is the first ViT-style backbone that explicitly enforces what–where separation throughout the architecture.

3 Method

3.1 What-Where Transformer: A Visual Backbone

Given an input image, WWT produces S pairs of representation vectors and spatial masks, designed to capture recognizable visual entities with spatial support, such as objects and background regions. We refer to each representation vector as a slot, following object-centric learning literature (Locatello et al., 2020). Here, S is a fixed number of slots, and $T = WH$ denotes the number of spatial tokens.

The network is designed to bind local patch tokens to slots to form meaningful groups during feed-forward inference, while contextualizing tokens with slots as their summaries. We therefore adopt

slot-token mutual communication as the basic building block of the architecture. A WWT block takes tokens, slots, and masks as inputs and outputs updated representations.

In the WWT block, we introduce a mutual attention layer (μAttn; Fig. 2), where tokens and slots interact via a dot-product attention mechanism. μAttn layer is defined using the token sequence $\boldsymbol{x}_i \in \mathbb{R}^{T\times d}$, the slot sequence $\boldsymbol{z}_i \in \mathbb{R}^{S\times d}$, and the mask $\boldsymbol{A}_i \in \mathbb{R}^{T\times S}$ from the i-th layer, as follows:

$$\boldsymbol{A}'_i = \boldsymbol{A}_i + \frac{Q(\boldsymbol{x}_i)K(\boldsymbol{z}_i)^\top}{\sqrt{d}}, \tag{1}$$

$$\boldsymbol{x}'_i = \boldsymbol{x}_i + \mathrm{Softmax}_S(\boldsymbol{A}'_i)V_1(\boldsymbol{z}_i), \tag{2}$$

$$\boldsymbol{z}'_i = \boldsymbol{z}_i + \mathrm{Softmax}_T(\boldsymbol{A}'^\top_i)V_2(\boldsymbol{x}_i), \tag{3}$$

where Q, K, V_1, and V_2 are learned projections, and $\mathrm{Softmax}_T$ and $\mathrm{Softmax}_S$ denote normalization along the token and slot dimensions, respectively. The token and slot updates use the same attention logits and are performed in parallel; this keeps the block's attention–MLP sequence close to that of a ViT block, without introducing additional sequential nonlinearities through an ordered update. Combining the propagated mask state with newly computed dot-product attention enables progressive refinement. For simplicity, the equations above describe single-head attention. In the multi-head setting with m heads, the token and slot dimensionalities remain unchanged, while the mask state has m channels: $\boldsymbol{A}_i \in \mathbb{R}^{m\times T\times S}$.

After μAttn, MLPs process tokens and slots separately. Tokens retain features tied to image coordinates, whereas slots aggregate more abstract, object-oriented information without fixed spatial coordinates. We therefore use independent MLPs rather than sharing a transformation. We also apply a point-wise MLP to refine the mask state. Before this MLP, we concatenate each token feature with its corresponding row of the mask state along the channel dimension. The output has S channels and is added to the token–slot attention logits in the next block:

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}'_i + \mathrm{MLP}_T(\boldsymbol{x}'_i), \tag{4}$$

$$\boldsymbol{z}_{i+1} = \boldsymbol{z}'_i + \mathrm{MLP}_S(\boldsymbol{z}'_i), \tag{5}$$

$$\boldsymbol{A}_{i+1} = \alpha\,\mathrm{LN}_A(\mathrm{MLP}_A(\mathrm{Concat}(\boldsymbol{A}'_i, \boldsymbol{x}'_i)))\,. \tag{6}$$

Here, MLP_T, MLP_S, and MLP_A have independent parameters. LN_A denotes layer normalization applied to the output of MLP_A, and α is a separate learnable scale applied after normalization. We adjust their hidden dimensions so that the three MLPs together have FLOPs comparable to the MLP in a ViT block. By processing the current attention state together with the corresponding patch features, MLP_A produces a patch-conditioned term that refines the token–slot attention logits in the next block. This allows patch-specific evidence, such as whether a patch is likely to belong to the background, to influence subsequent token–slot associations. We omit a residual connection around MLP_A because it led to training instability.

The WWT network is configured as a stack of WWT blocks. We omit spatial hierarchy or gradual downsampling for simplicity and to maintain the original-resolution attentions. In the first WWT block of the network, the initial tokens are obtained by applying a linear transformation to the RGB values of each patch. The initial slots are learnable queries initialized randomly and updated during training (Carion et al., 2020). The initial mask is given as zero-filled values. Positional embeddings are added only to the initial tokens. From the final L-th layer's outputs, $(\boldsymbol{z}_L, \boldsymbol{A}_L) = \mathbb{R}^{S\times(d+W\times H)}$ can be used as an image representation decomposed into S factors.

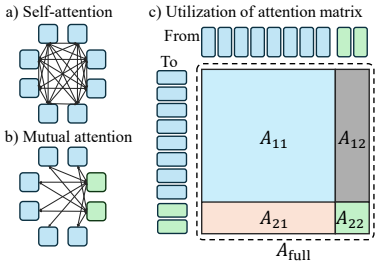


Figure 3: Illustration of ViT's token-token connectivity and WWT's token-slot connectivity.

Connectivity pattern WWT removes token-to-token and slot-to-slot interactions, using only bidirectional slot-token communication. Unlike self-attention, which mixes all tokens globally, WWT retains only the off-diagonal attention blocks A_{12} and A_{21} with a symmetry constraint $A_{21} = A_{12}^\top$ (Fig. 3c). This largely reduces computation and VRAM usage for high-resolution inputs, and is essential for efficiently implementing the attention pathway in WWT, which forwards A_{21} across layers. This connectivity bottleneck forces tokens to communicate only through slots, encouraging binding to informative slots for contextualization.

What-where decomposed representation WWT outputs slot-mask pairs rather than dense feature tensors or token-sequence representations, yielding an explicitly decomposed representa-

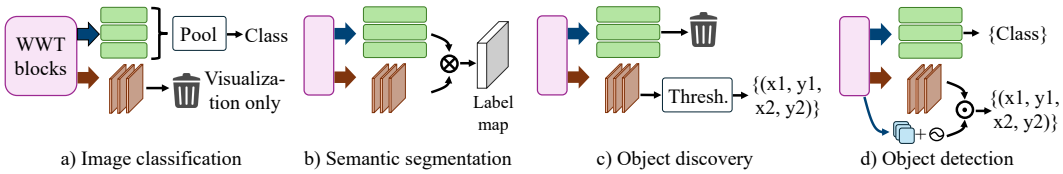


Figure 4: Task heads utilizing slot-mask representations from WWT (see Sec. 3.2).

tion. The slot-mask representation admits reconstruction into a dense feature tensor as $F_{hwd} = \sum_{s=1}^{S} A_{shw} \boldsymbol{z}_{sd}$, which can be interpreted as a learned factorization of dense feature maps into spatial assignment tensors and slot-wise feature bases. This formulation relates WWT to feature-factorization methods such as Deep Feature Factorization (DFF) (Collins et al., 2018; Fel et al., 2023), while differing in that WWT performs inference directly in the factorized latent space rather than decomposing features post hoc. From the perspective of what-where separation, conventional feature maps encode *what* in feature vectors while relying on spatial coordinates as hard-coded *where* representations. In contrast, WWT introduces explicit learnable *where* representations via slot-masks, potentially facilitating emergent grouping and non-local object-level interactions.

3.2 Task Heads Using Slots and Masks

By using the set of slot-mask pairs produced by WWT as an image representation, it can be applied to recognition and localization tasks with relatively simple task-specific heads as shown in Fig. 4.

Classification Head (a) The output slots $\boldsymbol{z}_L \in \mathbb{R}^{S \times d}$ are fed into a multilayer perceptron (MLP) and slot-wise C-class classification $\boldsymbol{c} \in \mathbb{R}^{S \times C}$ is obtained. After slot-wise classification, the results are averaged along the S dimension to obtain the global image class. Optionally, we can compute class activations (CA) for a predicted class p from the slot-classes and output masks by a class-confidence-weights average of masks as

$$\text{CA} = \frac{1}{S} \sum_{s=1}^{S} \boldsymbol{c}(s,p) \text{ReLU}(A_L(s)) \in \mathbb{R}_{\geq 0}^{W \times H}, \tag{7}$$

where $\boldsymbol{c}(s,p)$ denotes the predicted probability (or logit) of class p in slot s. The mask is allowed to have negative values, so we rectify them. This activation is used for visualization and interpretation.

Segmentation Head (b) The output masks $\boldsymbol{A}_L$ are reshaped into patch-level coarse region labels $\boldsymbol{M}_L \in \mathbb{R}^{W \times H \times S}$. If necessary, this is upsampled. This is then combined with per-slot classification results $\mathbb{R}^{C \times S}$ via matrix multiplication to obtain the final label map $\mathbb{R}^{C \times rW \times rH}$, where r denotes the upsampling factor. This head can be weakly supervised with image-level class labels by training the classification head and repurposing it, or supervised by dense pixel labels.

Object Discovery Head (c) In object discovery without additional training, object regions are obtained by binarizing each mask with a fixed threshold, extracting connected components, and removing duplicates. For single-object discovery, these are treated as candidates, and a post-processing step selects the most object-like region. Although heuristic, the region with the highest spatial concentration (Herfindahl, 1997) over the mask is selected.

Object Detection Head (d) For box-supervised detection, WWT predicts boxes from masks in a differentiable manner. To preserve spatial information, coordinate encodings (Liu et al., 2018) are concatenated to the patch tokens. The mask is used to softly aggregate the tokens and coordinates in the object region, and the aggregated feature is fed into MLPs for box regression, as we found the slot embedding itself, focusing on *what*, to be insufficient for box regression. Object classification, including the background class, is performed for each slot. Similar to DETR, bipartite matching is used during training to assign slots to ground-truth objects.

Autoencoding Head During ImageNet training, an auxiliary autoencoding head reconstructs the input RGB image using a small CNN. Its input feature map is formed as in the segmentation head, except that A_L is softmax-normalized along the slot axis before aggregation. The reconstruction loss encourages the masks $\boldsymbol{A}_L$ to group regions that can be reconstructed together. For a comparison with DINOSAUR (Seitzer et al., 2023), we also use this head in a distillation-based unsupervised object-centric learning setting, replacing RGB reconstruction with prediction of features from a frozen DINO backbone (Caron et al., 2021) using an MLP decoder.

Table 1: Classification accuracy and explainability on ImageNet-1k.

Model Size			Accuracy	Explainability	
Model	FLOPs	Params	Top-1 Acc.	Drop↓	Inc.↑
DeiT-Tiny	1.3G	5M	**72.2**%	52.6%	0.5%
+ Rollout	–	–	–	38.6%	4.4%
WWT-Tiny (ours)	1.3G	9M	71.3%	**30.1**%	**6.8**%
DeiT-Small	4.6G	22M	**79.8**%	57.5%	0.2%
WWT-Small (ours)	4.2G	31M	77.3%	**44.2**%	**4.5**%

Table 2: Weakly Supervised Semantic Segmentation Performance (mIoU) on ImageNet-S 50

Method	Activation	mIoU (%)
DeiT-Tiny	Raw CLS attention	8.8
DeiT-Tiny	Rollout (Abnar & Zuidema, 2020)	35.9
WWT-Tiny (ours)	Raw mask output	**49.1**
ResNet18 (He et al., 2016)	SEAM (Wang et al., 2020)	45.2
(reprod. by (Gao et al., 2022))	SC-CAM (Chang et al., 2020)	38.5
	AdvCAM (Lee et al., 2021)	46.9

3.3 Relationship with Existing Architecture

ViT WWT replaces ViT's token self-attention with bidirectional token–slot interactions. This design enables object-level representation with reduced complexity by decoupling patch length T and slot length S. Unlike the CLS token in standard ViT, WWT maintains multiple slots that, like registers (Darcet et al., 2024), persist across layers. Each slot is coupled to a spatial mask throughout the backbone, and these masks exhibit object-level grouping in our experiments.

DETR WWT shares the use of learned queries and cross-attention-like interactions with DETR, but embeds these mechanisms within the encoder. This design unifies feature extraction and localization without requiring a separate decoder, reinforcing a backbone-centric formulation for localization. In contrast to DETR, where objectness does not naturally emerge without strong supervision, even in self-supervised settings (Dai et al., 2021; Chen et al., 2023), WWT exhibits object-centric representations as an inherent property of the architecture.

Slot Attention Whereas conventional Slot Attention iteratively refines slots within a separate grouping module, WWT amortizes slot binding across backbone blocks in a single forward pass, jointly learning visual features and slot formation. While conventional Slot Attention typically aims for one slot per object, WWT does not require a one-to-one slot-object correspondence and permits an overcomplete set of slots to increase representational capacity. Our post-training slot reduction experiment further shows that WWT can adapt to object-centric grouping with fewer slots.

4 Experiments

We investigated the performance of WWT on various real-image tasks using zero-shot transfer and unsupervised or supervised finetuning after ImageNet-1k pretraining.

Implementation and training We follow the DeiT supervised ViT training recipe (Touvron et al., 2021) on ImageNet, with minor modifications. We train WWT-Tiny and WWT-Small, sized comparably to their DeiT counterparts, on ImageNet-1k with classification and autoencoding heads. We use label-smoothed cross-entropy ($p = 0.1$) (Szegedy et al., 2016) for classification and mean squared error on normalized RGB inputs for autoencoding, backpropagating both losses through slots and masks. We finetune the pretrained WWT-Small into WWT-OCL for unsupervised discovery and WWT-DET and WWT-SEG for supervised detection and segmentation, respectively. Further details appear in the Appendix.

Classification and explainability First, we verified whether the ImageNet supervised pretraining was successful based on the classification results in Table 1 and the explainability metrics. Despite replacing dense token-to-token self-attention with slot-mediated mutual attention, WWT retains substantial classification capability: its top-1 accuracy is 71.3% versus 72.2% for DeiT-Tiny, and 77.3% versus 79.8% for DeiT-Small. The widening gap at Small scale indicates a classification trade-off, which we examine alongside the localization benefits below. Furthermore, regarding the Drop and Increase (Inc.) metrics (Chattopadhay et al., 2018), which measure the validity of input attribution for classification results, WWT significantly outperforms DeiT when using class-token

Table 3: CorLoc (%) for zero-shot single object discovery

Model	Training	Post-processing	VOC12	COCO20k
DeiT (reprod. by (Darcet et al., 2024))	Supervised	LOST	11.7%	10.7%
DeiT + Reg. (Darcet et al., 2024)	Supervised	LOST	27.1%	25.1%
DeiT + LaSt (Shi et al., 2026) (our reprod.)	Supervised	Threshold	29.7%	23.1%
WWT (ours)	Supervised	Threshold	**41.4%**	**30.2%**
Obj-DINOv2 (Rawlekar et al., 2026)	Self-supervised	TokenCut	30.7%	19.7%
Obj-DINOv3 (Rawlekar et al., 2026)	Self-supervised	TokenCut	36.0%	23.4%
DINOv2 (reprod. by (Darcet et al., 2024))	Self-supervised	LOST	35.3%	26.9%
DINOv2 + Reg. (Darcet et al., 2024)	Self-supervised	LOST	55.4%	42.0%
DINOv1	Self-supervised	LOST	**61.9%**	**50.7%**

Table 4: Recall for zero-shot multi-object discovery using the top K predictions.

Dataset	Method	$K=1$	$K=5$	$K=10$	$K=15$
VOC12	DINOv1 + MOST (Rambhatla et al., 2023)	**18.9%**	34.0%	34.2%	33.5%
	WWT-Tiny (ours)	15.8%	**35.2%**	**38.3%**	**42.4%**
COCO20k	DINOv1 + MOST	**6.7%**	**12.4%**	13.0%	13.6%
	WWT-Tiny (ours)	4.8%	11.3%	**17.2%**	**19.4%**

attention maps or attention masks as they are. This advantage is not reversed even when the post-hoc interpretation method Rollout (Abnar & Zuidema, 2020) is applied only to ViT.

ImageNet weakly supervised segmentation To evaluate whether the attention maps capture object regions more precisely, we conducted a weakly supervised segmentation protocol using a classifier on the region-labeled subset ImageNet-S (Gao et al., 2022) 50. As shown in Table 2, WWT again demonstrated superior performance compared to DeiT and the baseline methods provided by the dataset authors. Notably, WWT outperformed convolutional ResNet (He et al., 2016), which is considered to have more local sensitivity than attentive models, combined with CAM-specific training methods (Wang et al., 2020; Chang et al., 2020; Lee et al., 2021).

Zero-shot object discovery Zero-shot object discovery (Siméoni et al., 2021; Rambhatla et al., 2023), which applies models to detection benchmarks without further training, was then conducted. Following the standard protocol, class labels were ignored, and evaluation was performed using CorLoc (Gokberk Cinbis et al., 2014), which does not penalize detections of non-annotated objects, as well as recall. While ViTs can usually perform object discovery via token clustering, e.g., LOST (Siméoni et al., 2021) and MOST (Rambhatla et al., 2023), WWT can be applied simply by thresholding its masks. Even in single-object discovery, WWT outperformed DeiT, including variants with registers, and achieved accuracy comparable to some of the strong self-supervised methods using DINOv2 (Oquab et al., 2024) and v3 (Table 3) [1].

In zero-shot multi-object discovery (Table 4), DINOv1-based MOST performs better when producing few predictions. As the number of predictions increases, WWT-Tiny surpasses MOST at $K = 5$ on VOC12 and $K = 10$ on COCO20k. MOST typically produced fewer than 10 predictions per image under its default clustering setting. To evaluate $K = 10$ and 15, we reran clustering at finer granularity to produce more candidates. Because these settings generate different prediction sets, recall need not increase monotonically with K; finer clustering may also fragment object regions. WWT-Tiny instead reaches 42.4% on VOC12 and 19.4% on COCO20k at $K = 15$, suggesting that its slot representations can support the discovery of multiple objects.

Unsupervised object discovery Many object-centric methods address unsupervised object discovery through slot-based autoencoding (Locatello et al., 2020; Seitzer et al., 2023; Kakogeorgiou et al., 2024). For comparison, we finetuned WWT under the same setting. WWT uses 64 backbone slots, far more than the six or seven slots typically used in SA-based models for VOC and COCO. During finetuning, reconstruction was applied only to the first six or seven slot-mask pairs, while the remaining slots were discarded. The results in Tab. 5 suggest that WWT performed mask-based discovery comparably with the SA-based baseline method, i.e., DINOSAUR, in the distillation-based setting. While some SA-based methods benefit from stronger autoregressive Transformer decoders, WWT enables object centric-

[1] DINOv1's strong performance in object discovery, unmatched by v2 and v3, has been reported in the literature (Darcet et al., 2024; Rawlekar et al., 2026).

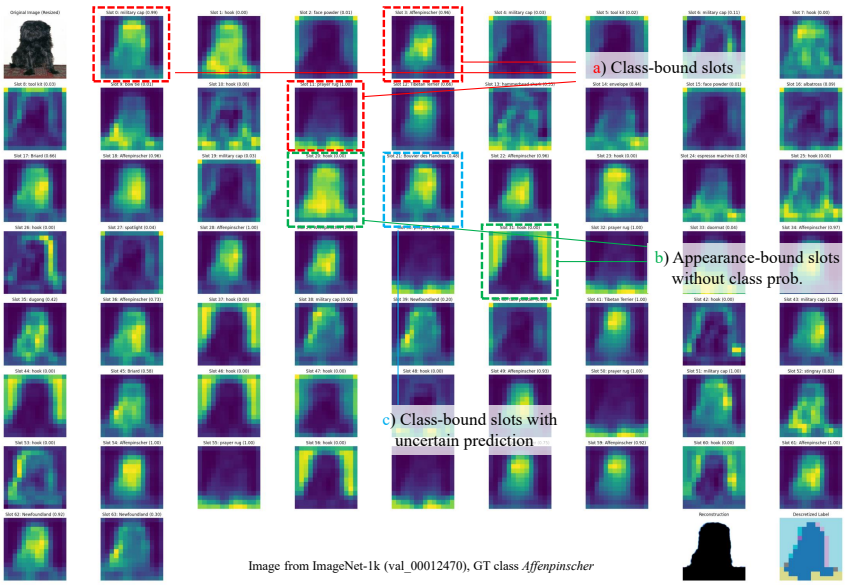


Figure 5: Visualization of per-slot output masks. Red (a): Class-bound slots attend to semantically meaningful regions and predict classes corresponding to the attended regions (with occasional prediction errors). Blue (b): Appearance-bound slots attend to regions but assign negligible class probabilities, e.g., *hook* with 0.0% top-1 class probability. Green (c): Class-bound slots with uncertainty attend to relevant regions but produce low-confidence predictions over confusing classes.

Table 5: Unsupervised object discovery results in the DINOSAUR (Seitzer et al., 2023) setting.

Method	Distil.	VOC12		COCO	
		mBO_i	mBO_c	mBO_i	mBO_c
WWT-OCL, zero-shot		29.9	31.8	16.2	19.5
WWT-OCL, RGB		22.8	25.3	15.5	16.3
WWT-OCL	✓	39.1 ± 0.2	**41.7 ± 0.4**	25.2 ± 0.3	29.0 ± 0.2
With MLP decoders					
Slot attention		24.6	24.9	17.2	19.2
DINOSAUR-MLP	✓	**39.5 ± 0.1**	40.9± 0.1	**27.7±0.2**	**30.9±0.2**
With autoregressive decoders					
SLATE		31.0 ± 0.4	32.4 ± 0.4	–	–
DINOSAUR-Trans.	✓	44.0 ± 0.8	51.2 ± 0.8	32.3±0.4	38.8±0.4

ity within the backbone itself. Its ability to perform this task without additional SA modules suggests that WWT internalized the representational bottleneck required for object binding. We also found that distillation from self-supervised dense representations is inevitable; RGB-reconstruction-based finetuning using the smaller datasets degraded the discovery performances from the zero-shot application of ImageNet-pretrained weights.

Analyzing what-where separation We moved synthetic patterns across four positions and compared the corresponding token and slot embeddings with those at Pos0 (Fig. 7). Slot embeddings remained similar, whereas token embeddings varied; the masks tracked the patterns' positions. These observations support a what-where separation in which slots encode translation-invariant *what* and masks represent spatial *where*. Additional examples appear in the Appendix.

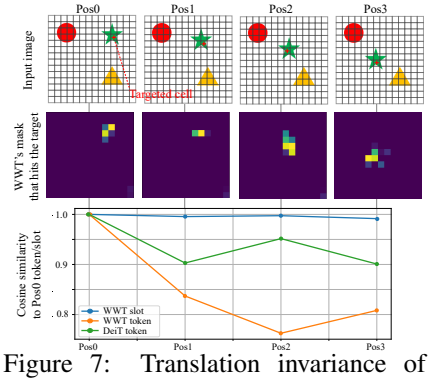


Figure 7: Translation invariance of patch tokens and slots.

Supervised detection and segmentation We finetuned WWT for detection and segmentation with lightweight heads in Fig. 4 (Table 7). Despite outnumbering slots, patch tokens yielded substantially lower detection performance (37.8 vs. 63.1 mAP), suggesting that localization information was more readily accessible from slots and masks. The same detection head on DeiT also per-

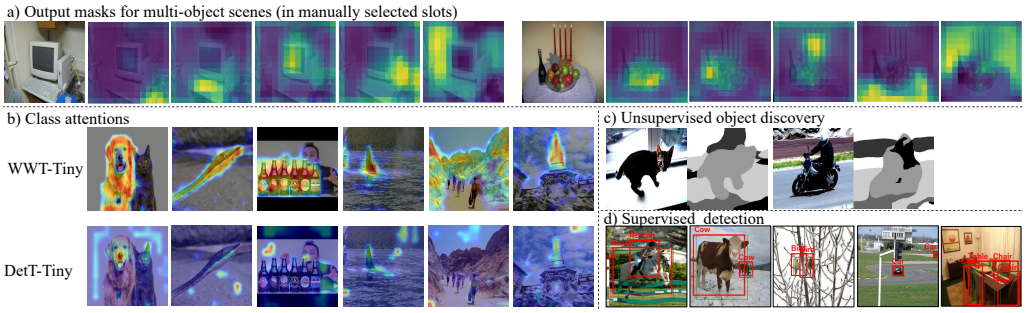


Figure 6: Visualization of the WWT's attentive masks and task results.

Table 7: Supervised results in detection and semantic segmentation in VOC

Detection in VOC07	mAP (%)	Segmentation in VOC12	mIoU (%)
WWT-DET	63.1	WWT-SEG	61.3
WWT + det. head on tokens	37.8	WWT + seg. head on tokens	59.9
DeiT + our det. head	56.4	DeiT + our seg. head	60.2
DETR	72.2	ResNet-50 FCN	66.5

formed worse (56.4 mAP), suggesting that the backbone's attention architecture matters for extracting localization from its internal representations. The slot-based head yielded a smaller gain in segmentation (61.3 vs. 59.9 mIoU). Unlike prior zero-shot and unsupervised localization methods with non-differentiable modules (Siméoni et al., 2021; Wang et al., 2022), WWT supports supervised finetuning with only minor modifications. Figure 6 shows qualitative results.

Hyperparameter and ablations Table 6 reports ablations under a preliminary augmentation setting, hence the differences from the main results. Among the tested slot counts, 64 gives the best results: more slots bring no gain, while fewer reduce accuracy. "W/o attention pathway" retains the bidirectional token-slot updates, slot count, and training objective, but recomputes attention logits independently at each block. This reduces CorLoc from 38.7 to 23.4, isolating the contribution of attention persistence within WWT. Sharing token and slot MLPs or removing the patch-conditioned attention term degrades both metrics. Removing the reconstruction loss barely affects Top-1 but reduces CorLoc to 19.8. As a further control, we apply the same reconstruction loss to DeiT, treating its tokens as reconstruction units in place of slots. For CorLoc, we select the map with the highest concentration measure, as for WWT; this variant achieves 25.7. These results indicate that WWT's localization performance depends on the combination of its architecture and the auxiliary reconstruction objective.

Table 6: Hyperparameter and component ablations of WWT-Tiny. All results use the ablation training configuration.

(a) Number of slots

Slots	Top-1	CorLoc
7	52.2	33.4
48	66.8	–
64	**69.7**	**38.7**
96	69.6	–
197	68.5	29.1

(b) Component ablations

Configuration	Top-1	CorLoc
Full model	**69.7**	**38.7**
W/o attention pathway	67.8	23.4
Shared token–slot MLP	62.3	34.5
W/o patch-conditioned term	68.8	31.4
W/o reconstruction loss	69.6	19.8
DeiT + reconstruction loss	71.5	25.7

5 Conclusion

We proposed a novel attentive vision architecture WWT, which decomposed the semantic and positional information into slots and masks. By introducing the inductive bias of what–where separation, WWT learned object-focused attention masks even under simple classification settings with RGB reconstruction. This design enabled the model to handle positional information as a distinct and generalizable component of image representation, with lightweight decoders for downstream tasks. We hope this work motivates further study of object-centric visual backbones and their potential for object grounding and spatial reasoning in future VLM/VLA systems.

Limitations Scaling WWT as effectively as ViT/DeiT remains an open challenge, as increasing model width yielded limited classification gains in our experiments. Moreover, our implementation uses a fixed number of slots, leaving adaptive slot allocation and its behavior in crowded scenes unexplored. Finally, while WWT provides an explicitly decomposed representation, applying its slot-mask pairs to established downstream tasks still requires task-specific adaptation. Exploring large-scale pretraining objectives and broader downstream applications is left for future work.

AI USE STATEMENT

We used generative AI tools for brainstorming, manuscript drafting, and generating experimental code. All AI-generated text and code were reviewed and verified by the authors.

REPRODUCIBILITY STATEMENT

The model architecture is described in Section 3. Training settings are detailed in Section 4 and Appendix A. We will release the training and evaluation code and model checkpoints upon acceptance.

REFERENCES

Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *Annual Meeting of the Association for Computational Linguistics*, pp. 4190–4197, 2020.

Sebastian Bock, Leonie Schüßler, Krishnakant Singh, Simone Schaub-Meyer, and Stefan Roth. MUFASA: A multi-layer framework for slot attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Guillem Brasó, Aljosa Osep, and Laura Leal-Taixé. Native segmentation vision transformers. *Advances in Neural Information Processing Systems*, 38:23299–23327, 2025.

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

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*, pp. 213–229, 2020.

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*, pp. 9650–9660, 2021.

Yu-Ting Chang, Qiaosong Wang, Wei-Chih Hung, Robinson Piramuthu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Weakly-supervised semantic segmentation via sub-category exploration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8991–9000, 2020.

Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. In *Winter Conference on Applications of Computer Vision*, pp. 839–847. IEEE, 2018.

Zeren Chen, Gengshi Huang, Wei Li, Jianing Teng, Kun Wang, Jing Shao, Chen Change Loy, and Lu Sheng. Siamese DETR. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15722–15731, 2023.

Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021.

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*, pp. 1290–1299, 2022.

Minkyu Choi, Kuan Han, Xiaokai Wang, Yizhen Zhang, and Zhongming Liu. A dual-stream neural network explains the functional segregation of dorsal and ventral visual pathways in human brains. In *Advances in Neural Information Processing Systems*, 2023.

Edo Collins, Radhakrishna Achanta, and Sabine Susstrunk. Deep feature factorization for concept discovery. In *European Conference on Computer Vision*, pp. 336–352, 2018.

Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. UP-DETR: Unsupervised pre-training for object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1601–1610, 2021.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations*, 2024.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pp. 7480–7512, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 248–255. Ieee, 2009.

Zhiwei Deng, Ting Chen, and Yang Li. Perceptual group tokenizer: Building perception with iterative grouping. *International Conference on Learning Representations*, 2024.

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. In *International Conference on Learning Representations*.

Mohammad K Ebrahimpour, Jiayun Li, Yen-Yun Yu, Jackson Reesee, Azadeh Moghtaderi, Ming-Hsuan Yang, and David C Noelle. Ventral-dorsal neural networks: object detection via selective attention. In *Winter Conference on Applications of Computer Vision*, pp. 986–994. IEEE, 2019.

Thomas Fel, Agustin Picard, Louis Bethune, Thibaut Boissin, David Vigouroux, Julien Colin, Rémi Cadène, and Thomas Serre. CRAFT: Concept recursive activation factorization for explainability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2711–2721, 2023.

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.

Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Multi-fold MIL training for weakly supervised object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2409–2416, 2014.

Anand Gopalakrishnan, Robert Csordás, Jürgen Schmidhuber, and Michael C Mozer. Decoupling the "what" and "where" with polar coordinate positional embeddings. *arXiv preprint arXiv:2509.10534*, 2025.

Anirudh Goyal and Yoshua Bengio. Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A*, 478(2266):20210068, 2022.

Karo Gregor and Yann LeCun. Emergence of complex-like cells in a temporal product network with local receptive fields. *arXiv preprint arXiv:1006.0448*, 2010.

Saurav Gupta, Sourav Lakhotia, Abhay Rawat, and Rahul Tallamraju. ViTOL: Vision transformer for weakly supervised object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 4101–4110, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Orris C Herfindahl. *Concentration in the steel industry*. Columbia university, 1997.

Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *International conference on artificial neural networks*, pp. 44–51. Springer, 2011.

Jindong Jiang, Fei Deng, Gautam Singh, and Sungjin Ahn. Object-centric slot diffusion. *Advances in Neural Information Processing Systems*, arXiv:2303.10834, 2023.

Ioannis Kakogeorgiou, Spyros Gidaris, Konstantinos Karantzalos, and Nikos Komodakis. SPOT: Self-training with patch-order permutation for object-centric learning with autoregressive transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22776–22786, 2024.

Tsung-Wei Ke, Sangwoo Mo, and Stella Yu. Learning hierarchical image segmentation for recognition and by recognition. In *International Conference on Learning Representations*, volume 2024, pp. 39881–39919, 2024.

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*, pp. 4015–4026, 2023.

Jungbeom Lee, Eunji Kim, and Sungroh Yoon. Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Liangzhi Li, Bowen Wang, Manisha Verma, Yuta Nakashima, Ryo Kawasaki, and Hajime Nagahara. Scouter: Slot attention-based classifier for explainable image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1046–1055, 2021.

Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *European Conference on Computer Vision*, pp. 280–296, 2022.

Yi Li, Hualiang Wang, Xinpeng Ding, Haonan Wang, and Xiaomeng Li. Token activation map to visually explain multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 48–58, 2025.

Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. Pay attention to mlps. *Advances in Neural Information Processing Systems*, 34:9204–9215, 2021a.

Kang Jun Liu, Masanori Suganuma, and Takayuki Okatani. Self-supervised learning of intertwined content and positional features for object detection. *International Conference on Machine Learning*, 267:39552–39567, 2025.

Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the CoordConv solution. *Advances in Neural Information Processing Systems*, 31, 2018.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021b.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986, 2022.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020.

Xu Ma, Yuqian Zhou, Huan Wang, Can Qin, Bin Sun, Chang Liu, and Yun Fu. Image as set of points. *International Conference on Learning Representations*, 2023.

A. David Milner and Melvyn A. Goodale. *The Visual Brain in Action*. Oxford University Press, Oxford, 1995.

Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection. In *European Conference on Computer Vision*, pp. 728–755. Springer, 2022.

Mortimer Mishkin, Leslie G Ungerleider, and Kathleen A Macko. Object vision and spatial vision: two cortical pathways. *Trends in neurosciences*, 6:414–417, 1983.

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. *Transactions on Machine Learning Research Journal*, 2024.

Namuk Park, Wonjae Kim, Byeongho Heo, Taekyung Kim, and Sangdoo Yun. What do self-supervised vision transformers learn? *International Conference on Learning Representations*, 2023.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

Bill Psomas, Ioannis Kakogeorgiou, Konstantinos Karantzalos, and Yannis Avrithis. Keep it SimPool: Who said supervised transformers suffer from attention deficit? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5350–5360, 2023.

Sai Saketh Rambhatla, Ishan Misra, Rama Chellappa, and Abhinav Shrivastava. MOST: Multiple object localization with self-supervised transformers for object discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15823–15834, 2023.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12179–12188, 2021.

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

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, 2017.

Maximilian Seitzer et al. Bridging the gap to real-world object-centric learning. *International Conference on Learning Representations*, 2023.

Cheng Shi, Yizhou Yu, and Sibei Yang. Vision transformers need more than registers. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Oriane Siméoni, Gilles Puy, Huy V Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. In *British Machine Vision Conference*, 2021.

Gautam Singh, Fei Deng, and Sungjin Ahn. Illiterate DALL-E learns to compose. In *International Conference on Learning Representations*, 2022.

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.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357, 2021.

Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. Resmlp: Feedforward networks for image classification with data-efficient training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):5314–5321, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Bowen Wang, Liangzhi Li, Yuta Nakashima, and Hajime Nagahara. Learning bottleneck concepts in image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10962–10971, 2023.

Bowen Wang, Liangzhi Li, Jiahao Zhang, Yuta Nakashima, and Hajime Nagahara. Explainable image recognition via enhanced slot-attention based classifier. *arXiv preprint arXiv:2407.05616*, 2024.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 568–578, 2021.

Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L Crowley, and Dominique Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14543–14553, 2022.

Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12275–12284, 2020.

Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. CvT: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22–31, 2021.

Yi-Fu Wu, Klaus Greff, Gamaleldin Fathy Elsayed, Michael Curtis Mozer, Thomas Kipf, and Sjoerd van Steenkiste. Inverted-attention transformers can learn object representations: Insights from slot attention. In *Causal Representation Learning Workshop at NeurIPS*, 2023.

Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, and Dan Xu. Multi-class token transformer for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4310–4319, 2022.

Timing Yang, Predrag Neskovic, Jansen Seheult, Wenchao Han, Anand Bhattad, Alan Yuille, and Feng Wang. RATS! patches talk through registers: Emergent parts in register attention transformers. *arXiv preprint arXiv:2606.14701*, 2026.

Ting Yao, Yehao Li, Yingwei Pan, Yu Wang, Xiao-Ping Zhang, and Tao Mei. Dual vision transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):10870–10882, 2023.

Stella X Yu, Ralph Gross, and Jianbo Shi. Concurrent object recognition and segmentation by graph partitioning. *Advances in Neural Information Processing Systems*, 15, 2002.

Weihao Yu, Chenyang Si, Pan Zhou, Mi Luo, Yichen Zhou, Jiashi Feng, Shuicheng Yan, and Xinchao Wang. Metaformer baselines for vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(2): 896–912, 2023.

Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. In *International Conference on Learning Representations*, 2023.

Junbo Zhao, Michael Mathieu, Ross Goroshin, and Yann Lecun. Stacked what-where auto-encoders. In *International Conference on Learning Representations Workshops*, 2016.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. *International Conference on Learning Representations*, 2021.

Table 8: Hyperparameters for WWT Training on ImageNet

	WWT-Tiny	WWT-Small
Depth	12	
MLP output dimensionality	192	384
Token MLP hidden dimensionality	768	576
Slot MLP hidden dimensionality	768	1536
Solver	AdamW	
Initial learning rate	3×10^{-4}	5×10^{-4}
Weight decay	0.05	
Mini-batch size	4096	
Training epochs	300	
Warm-up epochs	5	
Learning rate schedule	Cosine	
Clipgrad	1.0	
Input image size	224×224	
Patch size	16×16	
Label smoothing factor	0.1	
Mixup α	0.25	0.8
CutMix α	0.6	.95
Random Erasing	0	0.25
Dropout	0.0	
Stochastic Depth	0.0	
Color Jitter	0.4	
Classification loss	Cross-entropy	
Reconstruction loss	Mean squared error over normalized RGB	
Weight of reconstruction loss	0.2	0.2

APPENDIX

A IMPLEMENTATION DETAILS

We implemented two model variants with 12 blocks and 64 slots: WWT-Tiny with an embedding dimension of 192 and 3 attention heads, and WWT-Small with an embedding dimension of 384 and 6 attention heads. We used training code from PyTorch Image Models (timm) (Wightman, 2019) with modifications, using the PyTorch framework (Paszke et al., 2019). For computation, we used our institutional internal GPU cluster, where each node has four NVIDIA H100 GPUs. WWT-Tiny and WWT-Small were trained in 20 and 22 hours in the single-node training. Table. 8 summarizes the ImageNet training hyperparameters. We pre-trained the models on the ImageNet-1k training set with both a classification head and an autoencoding head attached. For weakly supervised semantic segmentation and zero-shot object discovery, segmentation and object discovery heads were additionally attached to the pre-trained model during inference. Since these heads contain no learnable parameters, no further training or fine-tuning was performed for these tasks. Table. 9 summarizes the finetuning hyperparameters.

B INFERENCE SPEED

Table 10 reports wall-clock inference latency and throughput. Despite comparable theoretical FLOPs, WWT has more individual operations across its token, slot, and attention pathways, which can increase kernel-launch overhead. WWT is therefore slower than DeiT at batch size 1, while their batched throughputs remain close.

B.1 PRELIMINARY RESULTS ON MODEL SCALING

We conducted preliminary experiments on scaling WWT toward DeiT-Base-class capacity. Table 11 compares width-only scaling (WWT-Base) with scaling both width and depth (WWT-Middle). In-

Table 9: Hyperparameters for WWT downstream finetuning

	WWT-OCL	WWT-SEG	WWT-DET
Solver		AdamW	
Initial learning rate	4×10^{-4}	6×10^{-4}	1×10^{-3}
Weight decay	1×10^{-5}	1×10^{-4}	1×10^{-5}
Mini-batch size	100	50	50
Training epochs		150	
Warm-up epochs		15	
Learning rate schedule		Cosine	
Clipgrad	2.0	1.0	0.8
Input image size	320	384	384
Patch size		16×16	
Label smoothing factor		Not used	
Color Jitter		0.4	
Weight of reconstruction loss	1.0	0	0

Table 10: Inference speed of WWT and DeiT. Latency is measured with batch size 1; throughput is measured with batch size 100.

	CPU		GPU	
Model	Latency (ms/img)	Throughput (img/s)	Latency (ms/img)	Throughput (img/s)
WWT-Tiny	35	41	6.7	6900
DeiT-Tiny	28	44	5.1	7200
WWT-Small	61	16	7.6	2900
DeiT-Small	54	18	6.2	3100

creasing width from WWT-Small to WWT-Base yielded only a slight Top-1 gain (77.3% to 77.4%), despite substantially higher compute and parameter counts. WWT-Middle achieved 77.9% with fewer FLOPs and parameters than WWT-Base. The classification gap to DeiT widened at larger model sizes. These results suggest that WWT may require a different scaling strategy from DeiT; its optimal scaling remains open.

Extended related work: Concurrent slot-mediated visual backbones. Concurrent work has also explored replacing direct patch-to-patch attention with communication through a smaller set of learned tokens. RATS (Yang et al., 2026) routes patch information through registers using compress-communicate-broadcast attention. With DINO-style self-supervised training, its registers exhibit consistent part-level structure without part annotations or an auxiliary part-level loss. VECA Song et al. (2026) uses learned cores as an interface between patch tokens, with an elastic core budget designed to improve the efficiency of high-resolution inference. It also reports semantically organized core responses and strong dense-prediction features. These studies and WWT share the observation that restricting patch communication through learned tokens can encourage spatially structured representations. WWT focuses on a different aspect of this design space: it maintains slot features and their patch-wise attention masks as coupled representations across backbone layers, and evaluates whether the resulting masks support localization and discovery after classification-oriented pretraining with auxiliary RGB reconstruction. The different training objectives and evaluation protocols preclude direct performance comparisons.

C Broader impact

We studied what-where separation as an inductive bias in visual backbones. Although our study focuses on a fundamental architectural principle rather than a deployment-ready application, we expect that our findings may facilitate future research in areas such as automation and safety, potentially leading to positive societal impact.

Table 11: Preliminary model scaling results on ImageNet-1k. WWT-Base scales width, whereas WWT-Middle scales both width and depth. FLOPs are measured at 224×224 input resolution.

Model	Dim.	Depth	FLOPs	Params	Top-1
DeiT-Small	384	12	4.6G	22M	79.8%
WWT-Small	384	12	4.2G	31M	77.3%
DeiT-Base	768	12	17.5G	86M	81.8%
WWT-Base	768	12	19.5G	142M	77.4%
WWT-Middle	576	16	15.8G	115M	77.9%

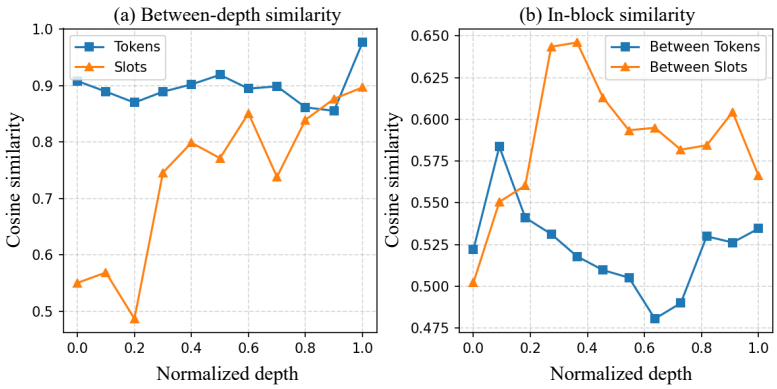


Figure 8: Depth-wise similarities of tokens and slots in and between depths in WWT.

At the same time, we are aware of potential misuse of image perception models, including bias amplification and privacy-invasive surveillance. We therefore encourage practitioners to carefully consider fairness and dataset bias before deployment in real-world systems.

D Additional analyses and visualization

Following the analysis of Park et al. (2023), we measured cosine similarity between tokens and between slots in WWT-Tiny to investigate evolution of tokens and slots. Figure 8(a) compares representations across depths: tokens change gradually, with similarity between adjacent blocks around 0.9, whereas slots change more substantially, particularly in early blocks. Figure 8(b) compares representations within each layer. Token diversity increases toward the middle of the network; slots initially become more similar, then regain moderate diversity. These patterns suggest that slots undergo substantial early reorganization while tokens remain more consistent across depth.

Figure 9 provides additional plots to supplement Fig. 7 using different shape patterns and real photographs, but the same target object was photoshopped into different positions. We observed consistently stronger translation-invariance in slots than in tokens. We note that WWT's tokens are taken from $\boldsymbol{x}_{L-1}$; the final token-MLP is omitted since it is not used for classification tasks and thus not pretrained.

Figures 10 to 15 are visualizations of all of the WWT's per-slot masks, paired with per-slot classification probabilities, aiming to complement Fig. 5 in the main text.

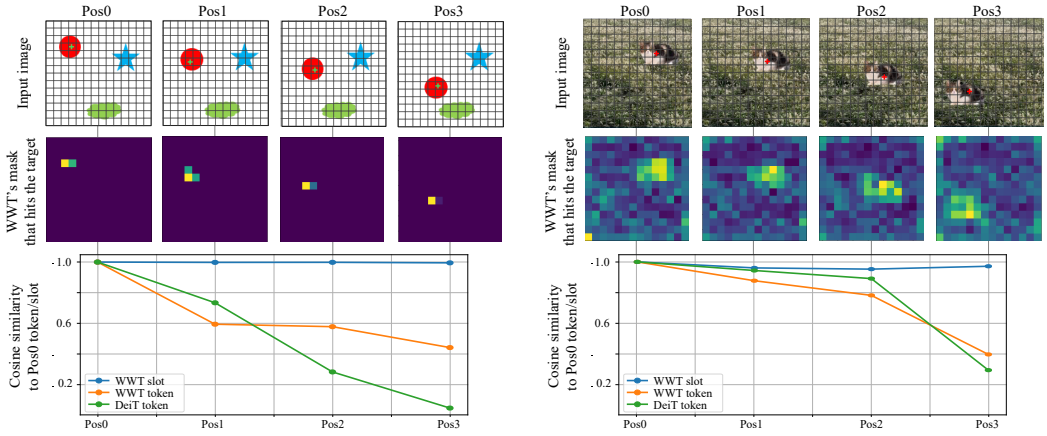


Figure 9: Translation invariance of patch tokens and slots with more synthesized images.

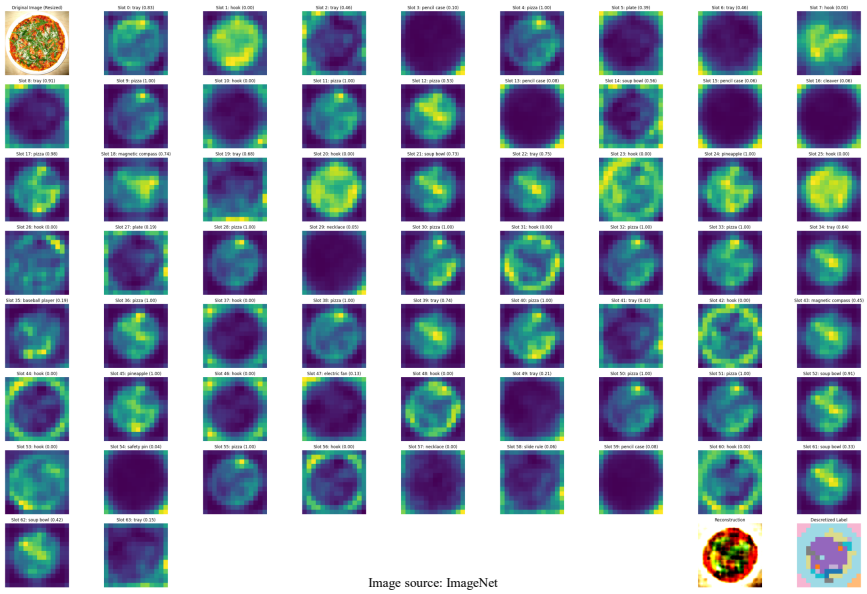


Figure 10: Visualization of per-slot attentive masks: Pizza.

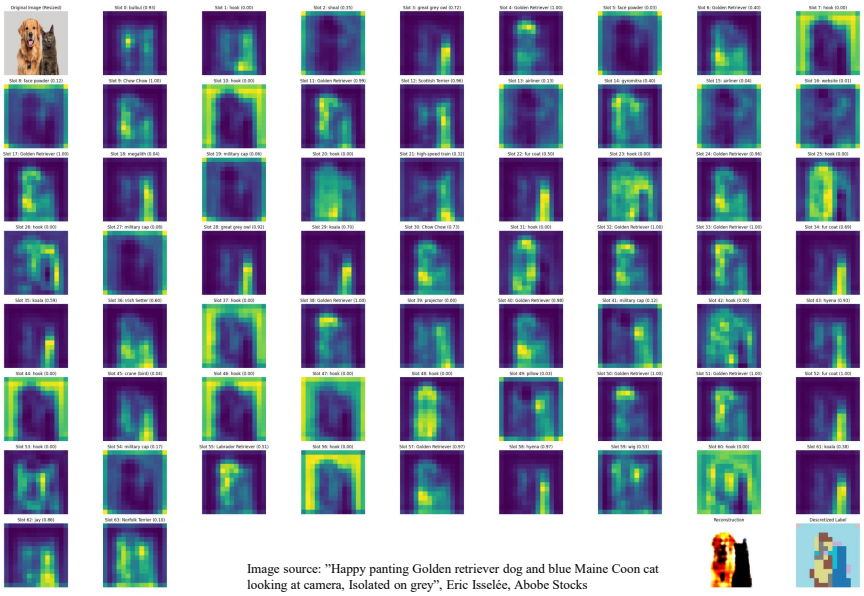


Figure 11: Visualization of per-slot attentive masks: Two-class objects, a dog and a cat.

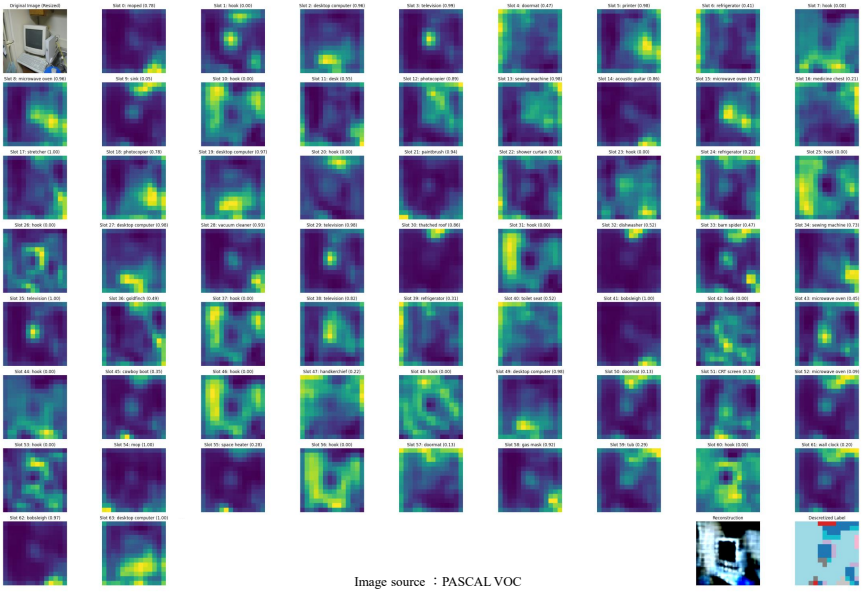


Figure 12: Visualization of per-slot attentive masks: Multi-object scene.

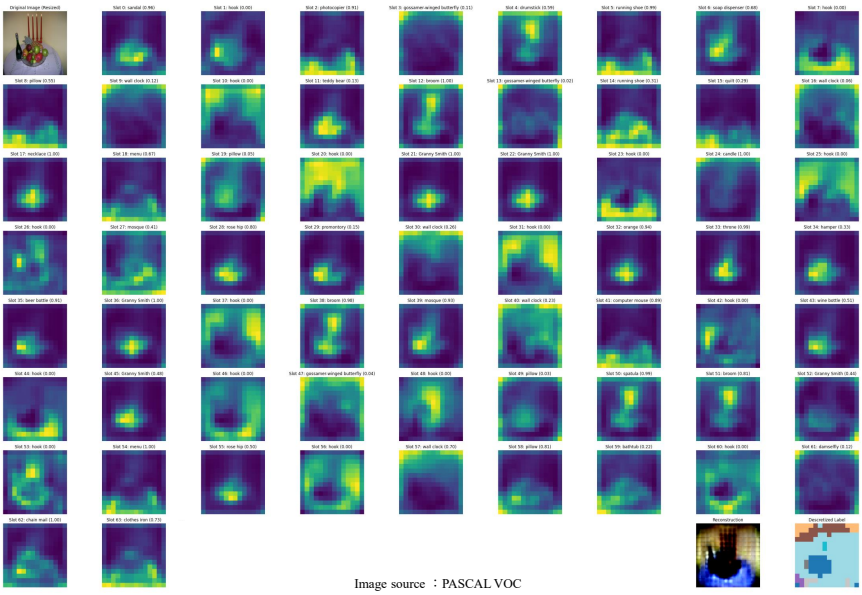


Figure 13: Visualization of per-slot attentive masks: Multi-object scene II.

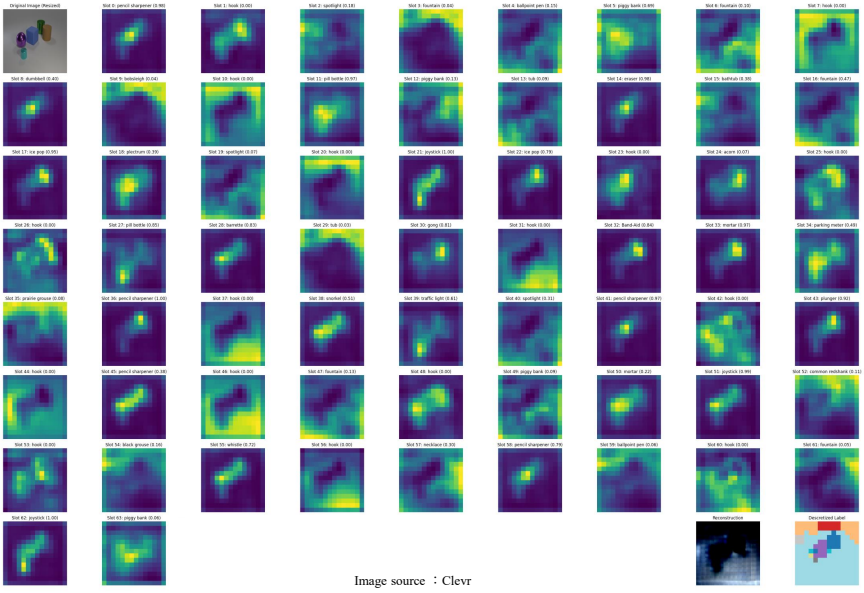


Figure 14: Visualization of per-slot attentive masks: Generalization from ImageNet to Clevr.

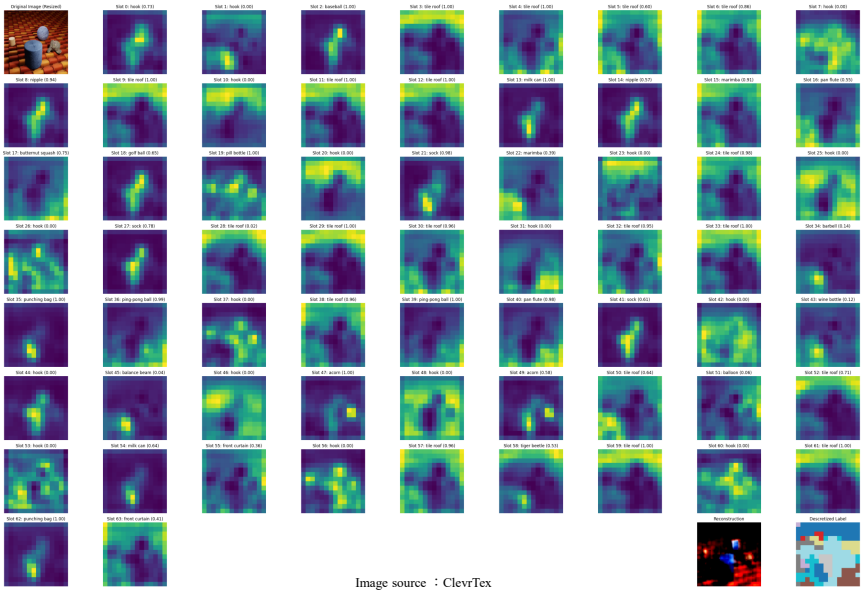


Figure 15: Visualization of per-slot attentive masks: Generalization from ImageNet to ClevrTex.