T-REN: Learning Text-Aligned Region Tokens Improves Dense Vision-Language Alignment and Scalability

Savya Khosla, Sethuraman T V, Aryan Chadha, Alex Schwing, and Derek Hoiem

University of Illinois Urbana-Champaign

Abstract. Despite recent progress, vision-language encoders struggle with two core limitations: (1) weak alignment between language and dense vision features, which hurts tasks like open-vocabulary semantic segmentation; and (2) high token counts for fine-grained visual representations, which limits scalability to long videos. This work addresses both limitations. We propose T-REN (Text-aligned Region Encoder Network), an efficient encoder that maps visual data to a compact set of text-aligned region-level representations (or region tokens). T-REN achieves this through a lightweight network added on top of a frozen vision backbone, trained to pool patch-level representations within each semantic region into region tokens and align them with region-level text annotations. With only 3.7% additional parameters compared to the vision-language backbone, this design yields substantially stronger dense cross-modal understanding while reducing the token count by orders of magnitude. Specifically, T-REN delivers +5.9 mIoU on ADE20K open-vocabulary segmentation, +18.4% recall on COCO object-level text-image retrieval, +15.6% recall on Ego4D video object localization, and +17.6% mIoU on VSPW video scene parsing, all while reducing token counts by more than 24× for images and 187× for videos compared to the patch-based vision-language backbone. The code and model are available at https://github.com/savya08/T-REN.

Keywords: Vision-language encoders · Region encoder network · Scalable representations

1 Introduction

Despite strong performance on global image-text tasks, modern vision-language encoders [3,10,23,30,38] remain bottlenecked by two structural limitations. First, cross-modal alignment between language and dense visual features remains weak, hindering open-vocabulary semantic segmentation, retrieval, and localization. Second, patch-based visual representations generate thousands of tokens per image, leading to substantial memory and compute overhead as visual input length increases. Together, these issues limit the fine-grained understanding and scalability of current vision-language systems.

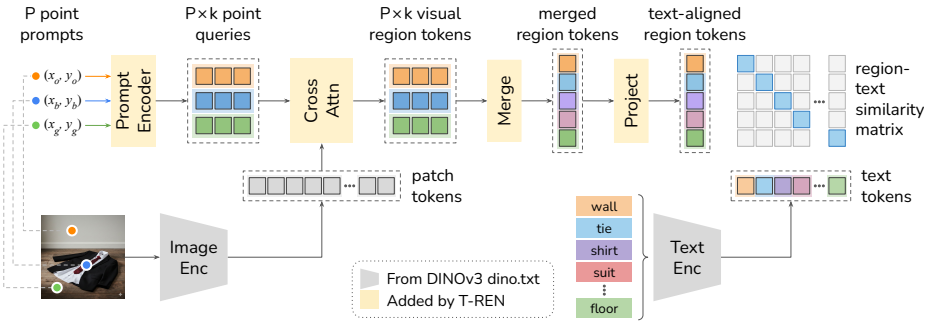


Fig. 1: Overview of T-REN. Given an input image and P point prompts, T-REN pools semantically related patch features via cross-attention to produce $k = 3$ region tokens per point prompt. Since multiple tokens may capture overlapping semantics (e.g., when points lie on the same region), highly similar tokens are merged to reduce redundancy. The resulting region tokens are projected into the text embedding space and matched with text encodings using cosine similarity for open-vocabulary tasks.

Existing approaches address these challenges in isolation. Token compression methods reduce the number of visual tokens but often sacrifice downstream performance [2, 19, 24, 37]. Conversely, methods that improve dense alignment continue to operate on patch-level representations [14, 15, 31, 40], inheriting the inefficiencies and semantic fragmentation of patch tokens. We argue that patch-level tokens are a suboptimal representational unit for dense vision-language modeling: they are too fine to be semantically meaningful, yet too numerous to scale efficiently.

We address both challenges simultaneously by shifting the unit of representation from patches to regions. We introduce T-REN, a Text-aligned Region Encoder Network that converts patch features from a vision backbone into a compact set of region tokens and aligns them with region-level text annotations. Jointly learning region-based pooling and region-text alignment produces stronger dense cross-modal representations while dramatically reducing the token count. We use DINOv3-based dino.txt [10] as the pretrained backbone.

Inspired by REN [11], T-REN uses a cross-attention module in which a grid of point prompts serves as queries over patch features from a frozen vision backbone. The attention operation pools patch features that correspond to the same semantic unit into region tokens. These region tokens are then projected into the text embedding space for alignment with region-level text annotations. Figure 1 provides an overview. Building on REN, we introduce a key refinement: instead of generating a single region token per point prompt, T-REN produces multiple tokens per prompt. This richer encoding captures both whole objects and their parts, alleviating the part-whole ambiguity inherent in REN and improving retrieval.

We evaluate T-REN on open-vocabulary semantic segmentation and retrieval over both images and videos. Our results show that T-REN consistently improves dense task performance while substantially reducing the number of visual tokens.

Importantly, these gains come at minimal cost, requiring only 3.7% additional parameters on top of the original patch-based vision-language backbone.

In summary, our contributions are:

1. **Improving cross-modal alignment.** By jointly learning the spatial pooling of patch features into region tokens and region-level text alignment, T-REN enhances dense vision-language understanding. Pooling and alignment associate text annotations with a precise visual region, enabling more accurate open-vocabulary segmentation (+5.9 mIoU on ADE20K, +15.8 on Cityscapes, and +17.6% on VSPW) and retrieval (+18.4% recall on COCO and +15.6% recall on Ego4D).
2. **Making visual encoding more scalable.** Adding only a small network on top of the pretrained backbone (+3.7% parameters), T-REN dramatically reduces visual token counts (e.g., 24.4× for COCO images, 187.5× for Ego4D videos, and 254.5× for VSPW videos). This enables the processing of long videos and large image collections, which would otherwise be computationally prohibitive.
3. **Enabling more expressive region encoding.** T-REN addresses the part-whole ambiguity present in its predecessor, REN. While REN generates a single region token per point prompt, T-REN produces multiple region tokens for each prompt, capturing both fine-grained object parts and the full object instance. This yields richer representations and improves retrieval (+5.7% recall on COCO).

2 Related Work

Vision-language encoders. The seminal CLIP [23] popularized large-scale contrastive image-text pretraining for open-vocabulary vision understanding. Several subsequent works have proposed improvements to this paradigm. For example, SigLIP [38] and SigLIP-2 [30] replace the contrastive objective with a sigmoid loss to improve pretraining efficiency, and PE [3] performs large-scale contrastive pretraining with joint image-video training to improve cross-modal alignment. In parallel, dino.txt [10] adapts powerful self-supervised vision backbones such as DINOv2 [21] and DINOv3 [28] for vision-language tasks, improving dense open-vocabulary understanding. T-REN builds on DINOv3-based dino.txt by introducing a lightweight region pooling and text-alignment module without retraining the underlying encoder.

Improving dense alignment. CLIP-based models are optimized for image-level understanding, leaving patch-level representations weakly aligned with language. To address this, training-free methods extract denser signals by modifying CLIP's attention at inference [14, 31, 40] or propagate spatial priors from strong vision models (e.g., DINO [6] or SAM [12]) into CLIP's feature space [15, 33]. In parallel, supervised methods enhance CLIP with region-level annotations through masked-region fine-tuning [18], region-text contrastive pretraining [22], or bounding-box alignment with detailed captions and hard negatives [35]. These

methods, however, operate on patch-level tokens, distributing object information across many tokens without semantic grouping. In contrast, T-REN pools patches into region tokens and jointly aligns them with text, producing semantically grounded representations.

Reducing the visual token count. Patch-based vision encoders produce hundreds or thousands of tokens per image, regardless of visual content, leading to substantial memory and compute overhead at scale. One line of work addresses this by pruning low-salience tokens at inference using predicted importance scores, attention weights, or diversity criteria [16, 24, 36, 37]. A complementary approach aggregates patch tokens using similarity-based pooling or cross-attention with a small set of learned query tokens [2, 9, 17, 19]. For video, temporally redundant tokens across frames are seldom merged to keep sequence length tractable [25, 29]. These methods generally trade a performance drop for efficiency. T-REN addresses this trade-off by producing tokens that are compact by construction: each region token pools an entire semantic unit, naturally yielding far fewer tokens without discarding spatial information.

Region-based representations. Shlapentokh-Rothman et al. [27] show that combining SAM [12] masks with DINOv2 [21] features produces compact and effective region representations for segmentation and retrieval, though the SAM segmentation step adds substantial overhead. REN [11] addresses SAM's cost with a lightweight cross-attention module that generates region tokens from point-prompt queries over frozen patch features, running 60× faster than SAM-based pipelines. However, REN generates only a single token per prompt, leading to part-whole ambiguity, and its tokens are not trained for text alignment. T-REN addresses these limitations by producing multiple tokens per prompt to capture hierarchical structure and jointly learning region pooling with text alignment using a DINOv3-based dino.txt backbone.

3 T-REN

Figure 1 provides an overview of T-REN. Our objective is two-fold: (1) improve fine-grained alignment between vision and language, and (2) reduce the token budget required to represent visual content. To achieve this, T-REN pools dense patch features from the DINOv3 ViT-L image encoder [28] into a compact set of semantically meaningful region tokens and aligns these tokens with text embeddings produced by the DINOv3-based dino.txt text encoder [10]. The pipeline and model architecture are described in Section 3.1, and the training objectives are detailed in Section 3.2.

3.1 Generating Compact Set of Text-Aligned Region Tokens

Encode point prompts into point queries. Following REN [11], T-REN employs point prompts to query and pool patch tokens into region tokens. Because a single spatial location may correspond to multiple semantic entities (e.g.,

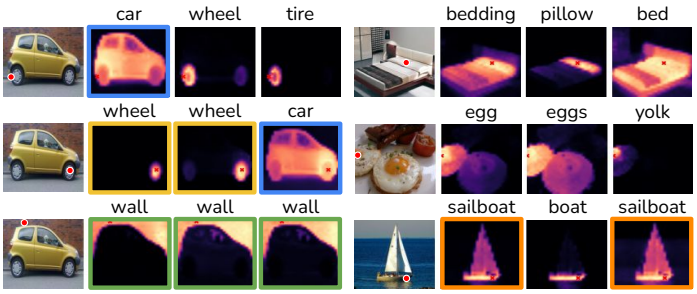


Fig. 2: Qualitative examples of T-REN's cross-attention masks. Each point prompt (shown in red) produces $k = 3$ cross-attention masks that pool patch tokens into region tokens. Tokens corresponding to masks covering the same semantic region are subsequently merged into a single region token. In the yellow-car example (left), masks highlighted with a blue border merge into a single *car* token, yellow masks into a *wheel* token, and green masks into a *wall* token.

overlapping objects or part–whole structures; see the blue prompt in Figure 1), we generate k region tokens per prompt. Our prompt encoder transforms each prompt (x_p, y_p) into $k = 3$ *point queries* using learned token embeddings. Specifically, the 2D position of the prompt is encoded with Gaussian Random Fourier Feature (RFF) embeddings and added to the learned tokens, yielding k distinct queries per location. To ensure full coverage of the image, we use a 2D grid of P prompts, resulting in $P \times k$ point queries in total.

Pool patch tokens into visual region tokens using point queries. The point queries are processed by a stack of $L = 2$ decoder layers. Each layer consists of two stages: (1) cross-attention between the point queries and the patch tokens, enabling each point query to gather spatially relevant visual information; and (2) self-attention among the k point queries associated with the same spatial location, allowing interactions among candidate entities that arise from a single prompt. To preserve spatial grounding across decoder layers, we re-inject the 2D positional encoding of each prompt after every self-attention block before passing the queries to the next layer. After L such layers, the processed queries are converted into *visual region tokens* using a final cross-attention operation over the patch tokens. This final cross-attention layer uses a single attention head and omits value and output projections, ensuring that the pooled features remain in the feature space of the frozen vision backbone. Due to the single-head design and our learning objective of pooling patch tokens within each region, the attention weight $a_i = \text{softmax}(q_i K^T / \sqrt{d_k})$ of a point query q_i naturally resembles a low-resolution region mask. Thus, each of the $P \times k$ point queries produces a *cross-attention mask* (Figure 2) that pools patch tokens into a corresponding region token, yielding $P \times k$ visual region tokens in total.

Merge similar visual region tokens. While a dense prompt grid ensures spatial coverage, it also introduces redundancy. For example, in Fig. 1, all point prompts placed on the wall will produce similar region tokens, as they correspond to the same semantic entity without any part-whole structure. To mitigate

this redundancy, we merge highly similar visual region tokens. Specifically, we average-pool tokens whose pairwise cosine similarity exceeds $\tau_{\text{token}} = 0.975$ or whose cross-attention mask IoU exceeds $\tau_{\text{mask}} = 0.8$. This simple merging strategy works because our training encourages tokens from the same semantic region to be identical (see Section 3.2 for more details). After merging, we obtain M *merged region tokens* representing the image, where M adapts to the visual complexity: visually sparse scenes yield fewer tokens, while cluttered scenes produce more tokens. We also cache the point-prompt coordinates of the merged region token, enabling association between region tokens and their spatial locations.

Align merged region tokens with text. Finally, we project the merged region tokens into the text embedding space using an MLP. The resulting text-aligned region tokens can be matched with text encodings via cosine similarity for open-vocabulary tasks.

Temporal aggregation for videos. For long videos, even a compressed set of per-frame region tokens results in an excessive number of tokens, especially for applications like episodic memory storage or streaming. We therefore extend our merging strategy temporally by aggregating region tokens across consecutive frames into *track tokens*. At each frame $t+1$, we compare its merged region tokens (prior to text projection) to the active track tokens accumulated up to frame t, where a track is considered active at time t if its object appears in frame t. We compute pairwise cosine similarity between the frame tokens and active track tokens and retain all pairs whose similarity exceeds $\tau_{\text{track}} = 0.65$. Then, greedy one-to-one matching is performed: each token from frame $t+1$ is assigned to the most similar active track. Tokens that do not match any active track initialize new tracks. This process is repeated sequentially across all frames in a streaming fashion, producing a set of object tracks. For each track, the constituent tokens are average pooled to obtain a single track token. We also cache the associated frame indices for each track token, preserving temporal grounding.

3.2 Training via Contrastive and Distillation Objectives

Our training objectives are designed around two complementary goals: (1) learning region tokens that capture object- and part-level semantics, and (2) preserving the rich representation space of the frozen vision-language backbone. To achieve the first goal, we apply contrastive losses in both the visual and text-aligned feature spaces. In the visual space, the contrastive objective (Eq. (1)) encourages region tokens originating from the same ground-truth mask to cluster together while pushing apart tokens from different regions, directly facilitating the similarity-based grouping that underlies our token merging strategy. In the text-aligned space, the contrastive objective (Eq. (2)) aligns region tokens with their corresponding category-level text encodings, improving region-level open-vocabulary recognition. To achieve the second goal, we apply distillation losses (Eq. (3)) that anchor both the visual and text-aligned region tokens to their respective targets obtained by mask-pooling features from the frozen backbone. Visual distillation prevents the learned tokens from drifting

away from the pretrained feature space, while text-aligned distillation preserves the open-vocabulary capability inherited from the vision-language backbone. Together, these four objectives ensure that region tokens are semantically coherent, groupable by similarity, and remain grounded in the backbone's representation space. We additionally supervise the cross-attention masks to match ground-truth masks (Eq. (4)), which we find accelerates convergence without affecting the final performance. Training uses a mixture of five segmentation datasets with biased point sampling to increase supervision from spatially overlapping regions, and Hungarian matching to handle the variable number of regions per prompt.

Data and Supervision Signal. T-REN is trained on a mixture of five segmentation datasets: COCOStuff [4], OpenImagesV7 [13], PhraseCut [32], Mapillary [20], and SA-1B [12]. For each training image, point prompts are sampled from locations inside ground-truth segmentation masks. To emphasize points that overlap multiple semantic entities (e.g., part-whole regions), sampling probability is proportional to the square of the number of overlapping regions. Distillation targets for visual region tokens are obtained by average pooling DINOv3 features within each corresponding mask. Targets for the text-aligned region tokens are obtained by processing the visual targets using the text-alignment vision block from DINOv3-based dino.txt [10, 28].

Hungarian matching. Each point prompt may have up to $k = 3$ target regions, while T-REN's cross-attention module always predicts k visual region tokens per prompt. To align predictions with target regions, we construct a cost matrix using cosine distances between predicted and target visual region tokens and solve a one-to-one assignment with the Hungarian algorithm [5]. Unmatched predicted tokens (when fewer than k targets exist) are excluded from the loss. This ensures permutation-invariant training and flexible assignment of predicted tokens to semantic entities.

Contrastive token learning. Using the Hungarian assignment, matched region tokens are supervised with contrastive objectives in both visual and text-aligned spaces. Formally, let $r_i^{(v)}$ and $r_i^{(t)}$ denote a normalized visual and text-aligned region tokens, m_i the corresponding region mask, and t_i the text encoding of the corresponding annotation. The contrastive losses for a batch of N regions are then computed as:

$$\mathcal{L}_{\text{cont}}^{(v)} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\sum_{j=1}^{N}\mathbb{1}_{[j\neq i,\ m_j=m_i]}e^{r_i^{(v)}\cdot r_j^{(v)}/\tau}}{\sum_{k=1}^{N}\mathbb{1}_{[k\neq i]}e^{r_i^{(v)}\cdot r_k^{(v)}/\tau}}, \tag{1}$$

$$\mathcal{L}_{\text{cont}}^{(t)} = -\frac{1}{2N}\sum_{i=1}^{N}\left(\log\frac{e^{r_i^{(t)}\cdot t_i/\tau}}{\sum_{k=1}^{N}\mathbb{1}_{[t_k\neq t_i]}e^{r_i^{(t)}\cdot t_k/\tau}} + \log\frac{e^{r_i^{(t)}\cdot t_i/\tau}}{\sum_{k=1}^{N}\mathbb{1}_{[t_k\neq t_i]}e^{r_k^{(t)}\cdot t_i/\tau}}\right). \tag{2}$$

Distillation loss. To keep region tokens aligned with the pretrained backbone, we apply a cosine-based distillation loss in both visual and text-aligned spaces,

encouraging predicted tokens to remain close to their Hungarian-assigned targets. Specifically, if $\tilde{r}_i^{(v)}$ and $\tilde{r}_i^{(t)}$ denote the visual and text-aligned targets,

$$\mathcal{L}_{\text{dist}} = \frac{1}{N}\sum_{i=1}^{N}\left[\left(1 - \frac{r_i^{(v)} \cdot \tilde{r}_i^{(v)}}{\|r_i^{(v)}\|\|\tilde{r}_i^{(v)}\|}\right) + \left(1 - \frac{r_i^{(t)} \cdot \tilde{r}_i^{(t)}}{\|r_i^{(t)}\|\|\tilde{r}_i^{(t)}\|}\right)\right]. \tag{3}$$

Attention supervision. Finally, to accelerate training, each max-normalized cross-attention mask $\tilde{a}_i$ is supervised to match the ground-truth masks m_i using a combination of binary cross-entropy loss ℓ_{bce} and DICE loss ℓ_{dice}:

$$\mathcal{L}_{\text{attn}} = \frac{1}{N}\sum_{i=1}^{N}\left[\ell_{\text{bce}}(\tilde{a}_i, m_i) + \ell_{\text{dice}}(\tilde{a}_i, m_i)\right]. \tag{4}$$

4 Experiments

We evaluate T-REN across three settings: zero-shot object-level retrieval over image databases (Section 4.1), open-vocabulary semantic segmentation (Section 4.2), and object localization and scene parsing in videos (Section 4.3). We do not evaluate on global image-level tasks (e.g., image classification or caption-based retrieval), as T-REN leaves the backbone's image-level representation unchanged. Consequently, for such tasks, T-REN preserves the original performance of the underlying DINOv3-based dino.txt model. Additional ablations and analyses are presented in Section 4.4. All experiments are conducted in a zero-shot manner.

4.1 Retrieval

We evaluate T-REN on the Visual Haystacks Single-Needle Challenge, where the task is: given a database of D images, answer queries of the form *"For the image containing the [anchor object], is there a [target object]?"*.

T-REN addresses this task using a simple two-step procedure. First, we retrieve the image containing the anchor object by computing the cosine similarity between the anchor text embedding and all text-aligned region tokens across the D images and selecting the image whose region token achieves the highest similarity score. Second, we determine whether the retrieved image contains the target object by computing the cosine similarity between the target text embedding and all region tokens within that image. We answer yes if the maximum similarity exceeds a threshold of $\tau_{\text{sim}} = 0.23$, and no otherwise.

The results are summarized in Table 1. T-REN consistently outperforms vision-language encoders (including patch-based models such as DINOv3 dino.txt and region-based models such as REN), as well as open-source MLLMs and retrieval-augmented methods across all values of D. It further surpasses Gemini-1.5 Pro and GPT-4o across all evaluated scales, while showing competitive performance with Gemini-3 Pro. Importantly, T-REN achieves these gains while incurring substantially lower computational costs than MLLMs.

Table 1: Visual Haystacks' single-needle challenge. T-REN outperforms open-source LMMs, RAG-based methods, and other vision-language encoders. "E" indicates context overflow, execution failure, or API error.

Model	D=1	D=2	D=3	D=5	D=10	D=20	D=50	D=100	D=500	D=1K
Detector Oracle	90.2	89.6	88.8	88.3	86.9	85.4	81.7	77.5	74.8	73.9
Proprietary LMMs										
Gemini-3 Pro	88.9	89.2	87.3	87.2	85.7	83.5	74.3	74.1	71.0	67.9
Gemini-1.5 Pro	88.4	82.0	78.3	76.0	71.9	68.6	62.8	57.4	E	E
GPT-4o	82.5	79.9	77.5	73.3	68.2	65.4	59.7	55.3	E	E
Open-source LMMs										
LongVILA	63.8	59.0	57.7	56.7	55.6	52.0	52.0	52.0	E	E
Qwen-2-VL	80.9	76.6	73.6	67.9	62.6	59.1	52.6	E	E	E
Phi-3	80.5	69.1	67.3	62.0	54.8	52.6	50.8	E	E	E
InternVL-2	88.1	80.5	72.3	63.9	58.8	55.2	E	E	E	E
mPLUG-OWL3	84.4	66.0	62.1	57.0	53.2	51.5	E	E	E	E
Retrieval-Augmented Methods										
LLaVA-v1.5	85.8	77.1	75.8	68.6	63.6	60.4	55.3	57.5	55.4	52.9
MIRAGE	83.2	77.8	76.6	72.8	70.5	66.0	63.6	62.0	58.7	55.7
Vision-Language Encoders										
SigLIP-2	72.0	69.2	68.1	65.3	64.1	60.3	58.7	58.3	56.6	54.9
REN	81.2	78.6	77.4	76.0	74.0	72.1	68.3	65.5	62.3	59.2
DINOv3 dino.txt	72.7	71.3	69.2	68.2	66.1	63.2	60.9	60.2	56.4	52.1
T-REN	**88.5**	**86.4**	**85.3**	**83.9**	**82.6**	**79.6**	**75.2**	**74.0**	**68.2**	**65.2**
$\Delta_{\text{vs. DINOv3 dino.txt}}$	**+15.8**	**+15.1**	**+16.1**	**+15.7**	**+16.5**	**+16.4**	**+14.3**	**+13.8**	**+11.8**	**+13.1**

In Figure 3, we analyze T-REN's performance on text-based retrieval (which is the first step in the Visual Haystacks approach). Compared to its vision-language backbone, T-REN improves recall (R@1) by an average of 18.4% across values of D. The performance gap widens as D increases. For example, at $D = 500$, the improvement reaches 28.6%. Notably, this gain is achieved while using 24.4× fewer tokens to represent the database on average across values of D.

4.2 Open-Vocabulary Semantic Segmentation

We evaluate T-REN on Open-Vocabulary Semantic Segmentation (OVSS), where the goal is to assign a semantic label to each pixel of an image in a zero-shot manner.

We prompt T-REN with a 24×24 grid of points and obtain $k = 3$ text-aligned region tokens per point. We then average-pool the k tokens at each point and compute cosine similarity against text encodings for all C classes in the dataset. This produces a $24 \times 24 \times C$ logit map, which is upsampled to the original image resolution of $384 \times 384 \times C$, and the most similar class is taken as the prediction for each pixel. To better assess fine-grained alignment at the point level, we disable token merging for this evaluation. This evaluation protocol follows the standard OVSS setup used to assess vision-language encoders [28].

The results are reported in Table 2. Compared to the DINOv3-based dino.txt, T-REN improves performance by +5.9 mIoU on ADE20k [39] and +15.8 mIoU on Cityscapes [7], demonstrating substantially stronger dense vision–language

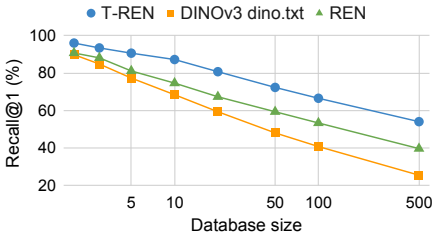


Fig. 3: Zero-shot retrieval. T-REN improves R@1 by 18.4% over DINOv3 dino.txt with 24× fewer tokens, and by 10.8% over REN, highlighting the advantage of multi-region token prediction.

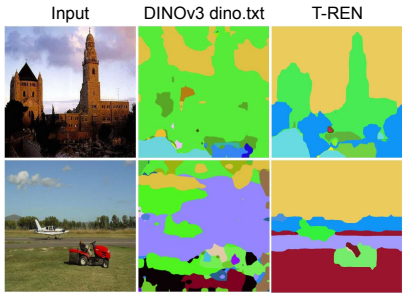


Fig. 4: Qualitative OVSS results. T-REN's segmentations better adhere to object boundaries than DINOv3 dino.txt's.

Table 2: Open-vocabulary semantic segmentation. T-REN outperforms both patch-based encoders and SAM-guided methods. T-REN and patch-based approaches use a ViT-L backbone with 576 tokens per image. We additionally report higher-resolution results (T-REN+) for a fairer comparison with SAM-guided methods, which use a ViT-H backbone and input resolutions of 672p and 1344p for ADE20k and Cityscapes, respectively.

Model	ADE20K	Cityscapes
SAM-guided approaches		
Trident	25.6	46.9
RADSeg+	29.9	45.8
TextRegion	27.3	47.4
Patch-based vision-language encoders		
CLIP	6.0	11.5
EVA-02-CLIP	10.9	14.1
SigLIP-2	10.8	16.3
PE	17.6	21.4
DINOv2 dino.txt	19.2	27.4
DINOv3 dino.txt	24.7	36.9
T-REN	**30.6**	**52.7**
T-REN+	**32.0**	**58.7**

alignment. As shown in Figure 4, T-REN's zero-shot segmentations adhere more closely to object boundaries, explaining the observed improvement. T-REN also surpasses recent SAM-guided OVSS approaches [1,26,34], which rely on an additional segmentation model (SAM [12]) for mask refinement and operate at higher input resolutions. Without any external refinement, T-REN already achieves superior performance at 384p. Furthermore, its accuracy improves consistently as input resolution increases (see Figure 6a).

4.3 Scaling to Video

Existing approaches to long-video tasks typically rely on either representing each frame with a single global token or maintaining dense patch-level representations while aggressively subsampling frames to control sequence length. Both strategies impose inherent trade-offs. Global frame tokens lack the spatial granularity needed to capture small objects in cluttered scenes, while temporal subsampling risks missing frames in which short-lived objects appear. In this section, we show that track tokens from T-REN provide an effective alternative: they preserve fine-grained spatial information by focusing on semantically meaningful regions while maintaining a compact token budget suitable for long video sequences. Consequently, T-REN yields consistent improvements in both performance and efficiency for retrieval and segmentation in the video setting.

Table 3: Video tasks. T-REN improves performance on both video query localization and video scene parsing while significantly compressing the representation. For query localization, following [8], a prediction is considered correct localization if the temporal-IoU between the predicted and target temporal window exceeds 0.25.

Model	Query Localization (Ego4D)			Scene Parsing (VSPW)	
	Recall@1	tAP	Compression (↑)	mIoU	Compression (↑)
DINOv3 dino.txt	36.8	14.4	1×	20.7	1×
REN	39.0	19.9	26.8×	18.5	22.9×
T-REN	**52.4**	**26.4**	**187.5×**	**38.3**	**254.5×**

Query localization in long videos. We evaluate T-REN on the task of localizing the last occurrence of an object in long episodic memory videos from Ego4D. Given a video and a query object, the goal is to identify the temporal window corresponding to the object's final appearance. The query is provided both as a text prompt and as a visual crop of the object. The videos average 140 seconds in duration and are sampled at 5 FPS. The target temporal window for the object's final occurrence spans 3 seconds on average.

To efficiently localize the queried object, we match video track tokens to the query by combining visual and textual similarity. Formally, we compute:

$$\texttt{track-similarity} = \texttt{visual-similarity} \times \texttt{textual-similarity},$$

where `visual-similarity` is the cosine similarity between the visual query tokens and the video's visual track tokens (i.e., track tokens obtained by temporally aggregating per-frame visual region tokens), and `textual-similarity` is the cosine similarity between the query text encoding and the video's text-aligned track tokens. We then retain all tracks whose similarity exceeds $\tau_{\text{sim}} = 0.18$ and report the temporal window of the last-ending track as the final prediction.

Table 3 summarizes the results. Compared to DINOv3-based dino.txt, T-REN improves query recall by 15.6% while using 187.5× fewer tokens to represent a video on average. This substantial reduction in token count yields important practical benefits. For example, in our evaluation with Ego4D, patch-based representations of long videos exceeded the memory capacity of a single NVIDIA A40 GPU and required streaming-based processing. On the other hand, T-REN, owing to its 187× compression, faces no such bottleneck: representations of even an 8-minute video (2400 frames) fit entirely within the memory of a single NVIDIA A40. These results highlight that T-REN is particularly well suited for episodic memory retrieval, where video representations must be stored efficiently on disk, fit within limited GPU memory, or be deployed on edge devices.

Video scene parsing. We evaluate T-REN on video scene parsing, where the goal is to assign a semantic label to every pixel in every frame of a video sequence.

To efficiently control the token budget for videos, we leverage temporally aggregated track tokens. Specifically, we compute the cosine similarity between each text-aligned track token and the text embeddings of all category labels, and assign the category with the highest similarity to the track. The predicted label for a track token is then applied to all spatio-temporal regions contributing

Table 4: Ablation on region pooling and text alignment. Removing either region pooling or text alignment leads to degraded performance, demonstrating that both components are essential for T-REN. "VH" denotes Visual Haystacks.

Train Region Pooling	Train Text Alignment	ADE20K (mIoU)	Cityscapes (mIoU)	VH Retrieval (D=10)	VH Reasoning (D=10)
		24.7	36.9	68.4	66.1
	✓	25.4	44.7	76.1	72.7
✓		19.5	21.1	65.5	68.8
✓	✓	**30.6**	**52.7**	**87.2**	**82.6**

to that track. Concretely, each track token is formed by first spatially merging point-prompted region tokens within a frame and then temporally aggregating similar tokens across frames; the assigned category is propagated to all spatial locations and time steps associated with that track.

Table 3 summarizes the results. T-REN surpasses both DINOv3-based dino.txt and REN while using 254.5× and 11.1× fewer tokens per video, respectively. We also analyze the effect of merging region tokens within and across frames (Table 5) and find that our merging strategy significantly reduces the token count without degrading representation quality (see Section 4.4 for more details).

4.4 Ablations

We perform ablations to validate our core design choices. First, we show that jointly learning spatial pooling and text alignment is critical for strong fine-grained vision–language alignment (Table 4). Next, we demonstrate that merging region tokens within and across frames removes redundancy in video representations without degrading quality (Table 5). We then highlight that multi-region token prediction is essential for learning expressive, hierarchically consistent region-based representations (Figure 5). In each of these studies, we modify only a single component while keeping the rest of the architecture and training protocol fixed. Finally, we analyze the impact of input resolution on T-REN and its generalization to classes unseen during training (Figure 6).

Ablating region pooling. We isolate the effect of region pooling by training a variant that bypasses spatial pooling and directly aligns patch-level features with region-level text annotations. Specifically, each patch token from the DINOv3 backbone is projected into the text embedding space and supervised using the annotation of the region in which it resides. As shown in Table 4, this variant underperforms T-REN, demonstrating that pooling patch tokens into region tokens not only leads to a reduced token count but also improves dense vision-language alignment.

Ablating text-alignment. We next evaluate the importance of jointly learning text alignment with region pooling. To this end, we train a variant that learns only region pooling and derives text-aligned region tokens post hoc. Specifically, we use the learned cross-attention masks to pool text-aligned patch features from the vision side of DINOv3-based dino.txt into region tokens. As shown in Table 4, this decoupled strategy yields substantially degraded performance. We

Table 5: Ablation on token merging strategies. In-frame merging alone achieves 29.2× compression with no mIoU loss; adding temporal merging reaches 254.5× compression at a cost of only 0.3 mIoU. Compression ratios are relative to a patch-based encoder baseline.

In-Frame Merging	Temporal Merging	VSPW mIoU	Comp. (↑)
		38.6	1×
✓		38.6	29.2×
✓	✓	**38.3**	**254.5**×

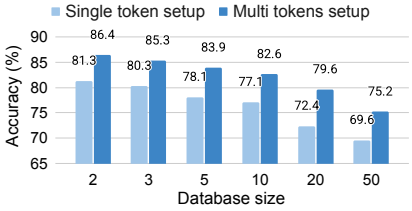


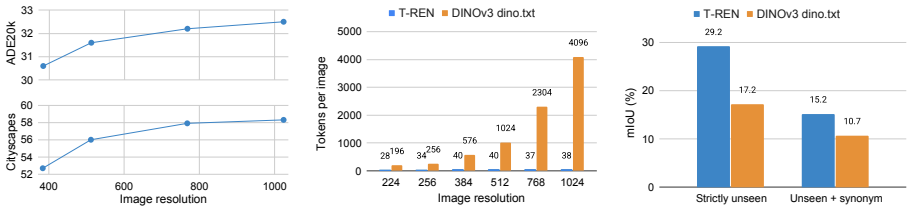
Fig. 5: Single vs. multi-token setups. Predicting multiple tokens per point consistently improves performance on Visual Haystacks, highlighting the benefit of capturing hierarchical visual structure.

attribute this to the spatial noise in independently learned text-aligned patch features, which often fail to respect object boundaries (see Figure 4). Consequently, although the region pooling module learns precise region assignments, applying them to spatially imprecise patch features produces misaligned semantic representations.

Ablating token merging. We analyze the impact of the proposed token merging stages, which consist of: (1) merging similar tokens produced by different point queries within a frame and (2) merging similar region tokens across frames. We measure their effect on the VSPW video scene parsing task, with results shown in Table 5. Both merging steps preserve task performance with negligible degradation while drastically reducing the number of tokens, indicating that the removed redundancy carries negligible discriminative information for this task.

Ablating multi-region token prediction per point prompt. A fundamental architectural difference between REN [11] and T-REN is that T-REN predicts multiple region tokens for each point prompt. To assess the impact of this design upgrade, we train a variant of T-REN that predicts only a single token per point prompt, keeping all other components unchanged. As shown in Figure 5, the single-token variant consistently underperforms the proposed multi-token setup in zero-shot retrieval and classification. This degradation indicates that constraining each point to a single token limits the model's ability to represent multiple valid hierarchical interpretations associated with a location (e.g., an object part and the full instance). Allowing multiple tokens per point preserves this part-whole structure and leads to more expressive region-level representations.

Effect of image resolution. Performance on vision tasks generally improves with higher input resolution, as illustrated for OVSS in Figure 6a. For patch-based encoders, however, the number of tokens scales quadratically with resolution, making high-resolution processing prohibitive for tasks such as image search (Section 4.1) and video query localization (Section 4.3), which require caching representations for large collections of images or frames. Although T-REN also relies on a patch-based backbone and therefore incurs higher computation when



(a) (b) (c)

Fig. 6: Resolution scaling and generalization. (a) Segmentation mIoU improves as input resolution increases. (b) Unlike patch-based encoders, token count of T-REN remains nearly constant as the image resolution is increased. (c) T-REN generalizes effectively to text classes unseen during training.

processing individual high-resolution images, it stores and propagates only aggregated region tokens for downstream tasks. This design keeps the number of cached tokens nearly constant regardless of the input resolution (see Figure 6b), allowing T-REN to benefit from higher resolution with minimal additional storage overhead.

Generalization to unseen categories. In Section 4.2, we evaluate T-REN on ADE20K, which is not used during training. However, T-REN is trained on a mixture of segmentation datasets containing over 4,600 category labels (Section 3.2), some of which overlap with the 150 ADE20K classes. To isolate true generalization, we identify five ADE20K categories that are entirely unseen during training (including synonyms): *conveyor belt*, *hovel*, *swivel chair*, *television receiver*, and *arcade machine*. We further identify 13 categories that appear in the training corpus only under different synonym forms. Evaluating performance on these subsets allows us to assess whether T-REN preserves the open-vocabulary capabilities of the underlying DINOv3 text encoder. As shown in Figure 6c, T-REN consistently outperforms DINOv3 dino.txt on these selected categories, mirroring its gains across the full 150-class benchmark.

5 Conclusion

We present T-REN, a vision-language encoder that learns text-aligned region tokens by jointly pooling patch features into region tokens and aligning them with language. This design produces compact representations while enabling fine-grained cross-modal grounding. As a result, T-REN supports both dense visual understanding and scalable representation for large visual collections. We demonstrate these advantages across diverse open-vocabulary settings, including image-level dense prediction, large-scale retrieval, and long-video parsing, supported by extensive ablations and analysis. Overall, T-REN shows that text-aligned region tokens provide a principled and scalable foundation for open-vocabulary vision-language modeling.

Future direction. While T-REN leverages strong pretrained vision-language backbones, future work may explore training region-based vision-language models end-to-end to further strengthen this paradigm.

References

1. Alama, O., Jariwala, D., Bhattacharya, A., Kim, S., Wang, W., Scherer, S.A.: Radseg: Unleashing parameter and compute efficient zero-shot open-vocabulary segmentation using agglomerative models. ArXiv (2025)
2. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your vit but faster. ArXiv (2022)
3. Bolya, D., Huang, P.Y., Sun, P., Cho, J.H., Madotto, A., Wei, C., Ma, T., Zhi, J., Rajasegaran, J., Rasheed, H.A., Wang, J., Monteiro, M., Xu, H., Dong, S., Ravi, N., Li, S.W., Dollár, P., Feichtenhofer, C.: Perception encoder: The best visual embeddings are not at the output of the network. ArXiv (2025)
4. Caesar, H., Uijlings, J.R.R., Ferrari, V.: Coco-stuff: Thing and stuff classes in context. CVPR (2016)
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers (2020)
6. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. ICCV (2021)
7. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
8. Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., Martin, M., Nagarajan, T., Radosavovic, I., Ramakrishnan, S.K., Ryan, F., Sharma, J., Wray, M., Xu, M., Xu, E.Z., Zhao, C., Bansal, S., Batra, D., Cartillier, V., Crane, S., Do, T., Doulaty, M., Erapalli, A., Feichtenhofer, C., Fragomeni, A., Fu, Q., Gebreselasie, A., González, C., Hillis, J., Huang, X., Huang, Y., Jia, W., Khoo, W., Kolář, J., Kottur, S., Kumar, A., Landini, F., Li, C., Li, Y., Li, Z., Mangalam, K., Modhugu, R., Munro, J., Murrell, T., Nishiyasu, T., Price, W., Puentes, P.R., Ramazanova, M., Sari, L., Somasundaram, K., Southerland, A., Sugano, Y., Tao, R., Vo, M., Wang, Y., Wu, X., Yagi, T., Zhao, Z., Zhu, Y., Arbeláez, P., Crandall, D., Damen, D., Farinella, G.M., Fuegen, C., Ghanem, B., Ithapu, V.K., Jawahar, C.V., Joo, H., Kitani, K., Li, H., Newcombe, R., Oliva, A., Park, H.S., Rehg, J.M., Sato, Y., Shi, J., Shou, M.Z., Torralba, A., Torresani, L., Yan, M., Malik, J.: Ego4d: Around the world in 3,000 hours of egocentric video. In: CVPR (2022)
9. Jaegle, A., Gimeno, F., Brock, A., Zisserman, A., Vinyals, O., Carreira, J.: Perceiver: General perception with iterative attention. ArXiv (2021)
10. Jose, C., Moutakanni, T., Kang, D., Baldassarre, F., Darcet, T., Xu, H., Li, S.W., Szafraniec, M., Ramamonjisoa, M., Oquab, M., Sim'eoni, O., Vo, H.V., Labatut, P., Bojanowski, P.: Dinov2 meets text: A unified framework for image- and pixel-level vision-language alignment. CVPR (2024)
11. Khosla, S., V, S.T., Lee, B., Schwing, A., Hoiem, D.: Ren: Fast and efficient region encodings from patch-based image encoders. In: NIPS (2025)
12. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.B.: Segment anything. ICCV (2023)
13. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., Duerig, T., Ferrari, V.: The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. IJCV (2020)

14. Lan, M., Chen, C., Ke, Y., Wang, X., Feng, L., Zhang, W.: Clearclip: Decomposing clip representations for dense vision-language inference. ArXiv (2024)
15. Lan, M., Chen, C., Ke, Y., Wang, X., Feng, L., Zhang, W.: Proxyclip: Proxy attention improves clip for open-vocabulary segmentation. In: European Conference on Computer Vision (2024)
16. Li, G., Liu, P.: Fastv-rag: Towards fast and fine-grained video qa with retrieval-augmented generation. ArXiv (2026)
17. Li, J., Li, D., Savarese, S., Hoi, S.C.H.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: ICML (2023)
18. Liang, F., Wu, B., Dai, X., Li, K., Zhao, Y., Zhang, H., Zhang, P., Vajda, P., Marculescu, D.: Open-vocabulary semantic segmentation with mask-adapted clip. ArXiv (2023)
19. Liang, Y., Ge, C., Tong, Z., Song, Y., Wang, J., Xie, P.: Not all patches are what you need: Expediting vision transformers via token reorganizations. ArXiv (2022)
20. Neuhold, G., Ollmann, T., Bulò, S.R., Kontschieder, P.: The mapillary vistas dataset for semantic understanding of street scenes. ICCV (2017)
21. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.Q., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.Y.B., Li, S.W., Misra, I., Rabbat, M.G., Sharma, V., Synnaeve, G., Xu, H., Jégou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision. TMLR (2023)
22. Pitawela, D., Carneiro, G., Chen, H.T.: Cloc: Contrastive learning for ordinal classification with multi-margin n-pair loss. CVPR (2025)
23. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: ICML (2021)
24. Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., Hsieh, C.J.: Dynamicvit: Efficient vision transformers with dynamic token sparsification (2021)
25. Shen, X., Xiong, Y., Zhao, C., Wu, L., Chen, J., Zhu, C., Liu, Z., Xiao, F., Varadarajan, B., Bordes, F., Liu, Z., Xu, H., Kim, H.J., Soran, B., Krishnamoorthi, R., Elhoseiny, M., Chandra, V.: Longvu: Spatiotemporal adaptive compression for long video-language understanding. ArXiv (2024)
26. Shi, Y., Dong, M., Xu, C.: Harnessing vision foundation models for high-performance, training-free open vocabulary segmentation. ArXiv (2024)
27. Shlapentokh-Rothman, M., Blume, A., Xiao, Y., Wu, Y., V, S.T., Tao, H., Lee, J.Y., Torres, W., Wang, Y.X., Hoiem, D.: Region-based representations revisited. In: CVPR (2024)
28. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., Massa, F., Haziza, D., Wehrstedt, L., Wang, J., Darcet, T., Moutakanni, T., Sentana, L., Roberts, C., Vedaldi, A., Tolan, J., Brandt, J., Couprie, C., Mairal, J., Jégou, H., Labatut, P., Bojanowski, P.: Dinov3. ArXiv (2025)
29. Tao, K., Qin, C., You, H., Sui, Y., Wang, H.: Dycoke : Dynamic compression of tokens for fast video large language models. CVPR (2024)
30. Tschannen, M., Gritsenko, A., Wang, X., Naeem, M.F., Alabdulmohsin, I., Parthasarathy, N., Evans, T., Beyer, L., Xia, Y., Mustafa, B., Hénaff, O., Harmsen, J., Steiner, A., Zhai, X.: Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. arXiv (2025)
31. Wang, F., Mei, J., Yuille, A.L.: Sclip: Rethinking self-attention for dense vision-language inference. ArXiv (2023)

32. Wu, C., Lin, Z., Cohen, S.D., Bui, T., Maji, S.: Phrasecut: Language-based image segmentation in the wild. CVPR (2020)
33. Wysocza'nska, M., Siméoni, O., Ramamonjisoa, M., Bursuc, A., Trzci'nski, T., P'erez, P.: Clip-dinoiser: Teaching clip a few dino tricks. In: ECCV (2023)
34. Xiao, Y., Fu, Q., Tao, H., Wu, Y., Zhu, Z., Hoiem, D.: Textregion: Text-aligned region tokens from frozen image-text models. ArXiv (2025)
35. Xie, C., Wang, B., Kong, F., Li, J., Liang, D., Zhang, G., Leng, D., Yin, Y.: Fg-clip: Fine-grained visual and textual alignment. ArXiv (2025)
36. Xing, L., Huang, Q., wen Dong, X., Lu, J., Zhang, P., Zang, Y., Cao, Y., He, C., Wang, J., Wu, F., Lin, D.: Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. ArXiv (2024)
37. Yin, H., Vahdat, A., Alvarez, J., Mallya, A., Kautz, J., Molchanov, P.: A-vit: Adaptive tokens for efficient vision transformer (2022)
38. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. ICCV (2023)
39. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. IJCV (2016)
40. Zhou, C., Loy, C.C., Dai, B.: Extract free dense labels from clip. In: ECCV (2021)