ULTra: Unveiling Latent Token Interpretability in Transformer-Based Understanding and Segmentation

Anonymous authors
Paper under double-blind review

Abstract

Transformers have revolutionized Computer Vision (CV) through self-attention mechanisms. However, their complexity makes latent token representations difficult to interpret. We introduce `ULTra`, a framework for interpreting Transformer embeddings and uncovering meaningful semantic patterns within them. `ULTra` enables unsupervised semantic segmentation using pre-trained models without requiring fine-tuning. Additionally, we propose a self-supervised training approach that refines segmentation performance by learning an external transformation matrix without modifying the underlying model. Our method achieves state-of-the-art performance in unsupervised semantic segmentation, outperforming existing segmentation methods. Furthermore, we validate `ULTra` for model interpretation on both synthetic and real-world scenarios, including Object Selection and interpretable text summarization using LLMs, demonstrating its broad applicability in explaining the semantic structure of latent token representations.

1 Introduction

In recent years, the Transformer architecture and foundation models, leveraging self-attention mechanisms to capture complex dependencies in text, have transformed Natural Language Processing (NLP) benchmarks (Vaswani et al., 2017; Touvron et al., 2023; Team et al., 2024). Similarly, Vision Transformers (ViTs) (Dosovitskiy et al., 2020) have been adapted in Computer Vision (CV) and now serve as the backbone for various tasks such as segmentation and object detection (Thisanke et al., 2023; Liu et al., 2021). Despite their success, understanding the interpretability of Transformers remains a challenge due to the complexity of their latent token representations.

Several methods have been developed to enhance the interpretability of CNN-based models (Simonyan et al., 2014; Zeiler & Fergus, 2014; Selvaraju et al., 2017). While some of these can be extended to Transformer architectures, they do not fully leverage the unique attention mechanisms inherent to Transformers. Recent research has introduced interpretability methods specifically designed for Transformers (Chefer et al., 2021b; Abnar & Zuidema, 2020; Vig & Belinkov, 2019). However, these approaches primarily focus on explaining final model outputs, providing limited insight into the intermediate processes that lead to predictions.

In this paper, we introduce ***U**nveiling **L**atent **T**oken Interpretability in T**ra**nsformer-Based Understanding (`ULTra`)* a framework for interpreting latent tokens in Transformers. Recent work with objectives similar to ours (Chen et al., 2024) attempts to interpret latent tokens by mapping them into CLIP's multi-modal embedding space, using an external text encoder to associate tokens with semantic descriptions. In contrast, our method directly analyzes the latent space of ViTs, uncovering the role and function of individual tokens within the high-dimensional representation without relying on auxiliary models. `ULTra` provides deeper insights into the internal representations of these models. By analyzing the semantic information captured by latent tokens, we show that Transformers inherently encode the semantic structure of their input as a collection of distinct concepts. Moreover, by aggregating explanation maps, `ULTra` achieves unsupervised semantic segmentation. Unlike prior unsupervised segmentation methods that require additional training (Sick et al., 2024; Hamilton et al., 2022; Li et al., 2023), `ULTra` leverages the intrinsic knowledge of pre-trained models on tasks other than semantic segmentation to achieve state-of-the-art performance on benchmark

datasets without the need for fine-tuning. To further enhance segmentation performance, we introduce a self-consistency approach that learns an external transformation matrix in a self-supervised manner, refining segmentation without modifying the underlying model. Additionally, we validate our interpretability framework on transformer-based LLMs through qualitative analyses in text summarization, demonstrating its broad applicability across modalities.

Our main contributions are as follows:

- We propose a framework for interpreting latent tokens in Transformers, uncovering the stored semantic knowledge in each latent token. To the best of our knowledge, we are the first to investigate latent token interpretability directly.
- We extend `ULTra` to several designed synthetic and real-world tasks, showcasing its versatility and applicability, including Object Selection and interpretation of LLMs in text summarization tasks.
- By aggregating explanation maps generated from latent tokens, our method enables unsupervised semantic segmentation using pre-trained ViTs without requiring any additional training. This strategy outperforms existing state-of-the-art approaches that rely on fine-tuning or supervision, highlighting the effectiveness of leveraging the semantic structure embedded in large-scale models.
- To further enhance segmentation performance, we introduce $\text{ULTra}_{\mathcal{W}}$, a lightweight learnable extension that optimizes a self-consistency loss. This loss encourages stable token representations under input perturbations and yields a transformation matrix that projects tokens onto a more informative subspace while keeping the backbone model unchanged.

2 Related Work

Interpretable Deep Learning. Model interpretability is a critical aspect of deep learning, particularly for complex architectures like Transformers. Traditional methods such as saliency maps (Simonyan et al., 2014), Grad-CAM (Selvaraju et al., 2017), LIME (Ribeiro et al., 2016), and SHAP (Lundberg, 2017) have been effective for CNNs but do not fully exploit Transformers' self-attention mechanisms. A growing body of research focuses on Transformer-specific interpretability techniques that leverage attention mechanisms as intrinsic explanations (Vig & Belinkov, 2019; Abnar & Zuidema, 2020; Chefer et al., 2021b; Jain & Wallace, 2019; Wu et al., 2024). These methods are largely post-hoc, providing explanations after model training. In contrast, ante-hoc approaches such as IA-ViT (Qiang et al., 2023) and Mechanistic Interpretability (Rai et al., 2024) seek to make Transformers inherently interpretable. Additionally, techniques like LeGrad (Bousselham et al., 2024) and IA-RED^2 (Pan et al., 2021) improve interpretability by analyzing feature formation and reducing redundancy in self-attention. Despite these advancements for model interpretability, understanding latent tokens in ViTs remains underexplored, highlighting the need for methods that explicitly interpret latent tokens. A related study (Chen et al., 2024) explores latent token interpretation in CLIP by modifying self-attention mechanisms.

Semantic Segmentation. Unsupervised semantic segmentation has progressed through self-supervised learning and clustering techniques. Early methods such as IIC Ji et al. (2019) and PiCIE Cho et al. (2021) leveraged mutual information and consistency principles to enhance feature representations. Transformer-based approaches like DINO Caron et al. (2021) and STEGO Hamilton et al. (2022) further improved segmentation by capturing meaningful structures through self-attention. Other techniques, including MaskContrast Van Gansbeke et al. (2021), Leopart Ziegler & Asano (2022), and ACSeg Li et al. (2023), refined segmentation through clustering and adaptive conceptualization. More recent methods, such as DepthG Sick et al. (2024), incorporate depth-guided correlations, while U2SEG Niu et al. (2024) utilizes pseudo-labeling. Additionally, SmooSeg Lan et al. (2023) enforces smoothness priors, and HSG Ke et al. (2022) applies hierarchical segmentation via multiview clustering Transformers. Moreover, in the context of semantic segmentation, Weakly Supervised Semantic Segmentation (WSSS) aims to generate segmentation masks using only image-level labels. These methods often rely on saliency maps from interpretability techniques, such as Class Activation Maps (CAMs), to localize objects Choe & Shim (2019); Yin et al. (2020); Chen et al.

(2023). Typically, they depend on class logits to guide segmentation, refining masks based on predicted class scores.

Latent Embedding. The high dimensionality and complex distribution of latent embeddings in deep models pose significant challenges for interpretation and manipulation. Methods like GroupViTXu et al. (2022) utilize hierarchical grouping to facilitate more meaningful representation learning, enabling semantic segmentation. Other approaches, such as Lee et al. (2024); Bolya et al. (2023); Liang et al. (2022), improve computational efficiency by eliminating redundant tokens, while register-based ViTsDarcet et al. (2024) address artifacts in feature maps caused by outlier-norm tokens through the introduction of register tokens. However, these techniques primarily emphasize computational efficiency and representation structuring rather than the interpretability of latent embeddings.

3 Methodology

In this section, we introduce our approach for interpreting latent representations in Transformers. We start by outlining the essential preliminaries, followed by a detailed explanation of the `ULTra` framework for analyzing latent tokens. Additional details on the Transformer architecture are provided in Appendix F.

3.1 Preliminaries

Previous research on attention-based model interpretability (Abnar & Zuidema, 2020; Chefer et al., 2021b;a; Wu et al., 2024) has primarily focused on analyzing the semantic flow from input tokens to class logits. This is typically achieved by adding the attention probability matrix of each layer to an identity matrix and aggregating the results across attention heads through a weighted average. The weights are derived from the gradients of the class logits with respect to the attention probabilities. Specifically, to assess the contribution of each class logit using attention information, the contribution map for layer b is defined as:

$$\mathbf{C}_c^{(b)} = \mathbf{I} + \mathbb{E}_h \left[\left(\nabla_{\mathbf{A}_h^{(b)}} p(c) \right)^+ \odot \mathbf{A}_h^{(b)} \right], \tag{1}$$

$$\overline{S}_c = \mathbf{C}_c^{(1)} \cdot \mathbf{C}_c^{(2)} \cdots \mathbf{C}_c^{(L)}, \tag{2}$$

$$S_c = \overline{S}_c[0, 1:], \tag{3}$$

where $\odot$ denotes the Hadamard product, and $(\cdot)^+$ is the operator that retains only positive values. $\mathbf{I}$ represent the identity matrix which reflect the contributions of skip connections and $\mathbf{A}_h^{(b)} \in \mathbb{R}^{(n+1)\times(n+1)}$ denotes the attention probability matrix of head h and layer b. The term $\nabla_{\mathbf{A}_h^{(b)}} p(c) = \frac{\partial p(c)}{\partial \mathbf{A}_h^{(b)}}$ is the partial derivative of the attention map with respect to the predicted probability for class c, and $\mathbb{E}_h$ denotes the mean over multiple attention heads. Equation 2 represents matrix multiplication, which accounts for aggregating contributions from all layers. Each element of $\overline{S}_c[i, j]$ represents the influence of the j-th input token on the i-th output token. Thus, each element of $S_c[j]$ quantifies the influence of the j-th input token on class c, with token 0 corresponding to the `CLS` token. This raises a new question: ***Is there a similar approach to interpret latent tokens in Transformers?***

3.2 ULTra Framework

Inspired by Equation 1, in this work we aim to measure the contribution of latent token $\mathbf{z}_i^{(l)}$ using the underlying attention probabilities, where $i \in \{0, 1, \ldots, n\}$ denotes the token index and l represents the layer. Specifically, to quantify the influence of the attention map at layer b on the latent token $\mathbf{z}_i^{(l)}$, the contribution map $\mathbf{C}_i^{(b,l)}$ is defined as follows:

$$\mathbf{C}_i^{(b,l)} = \mathbf{I} + \mathbb{E}_h \left(\left(\nabla_{\mathbf{A}_h^b} f(\mathbf{z}_i^{(l)}) \right)^+ \odot \mathbf{A}_h^b \right), \tag{4}$$

Intuitively, this approach traces the most influential input information contributing to a latent token $\mathbf{z}_i^{(l)}$, where the notion of *influence* is defined based on the impact an input token has on the function $f : \mathbb{R}^n \to \mathbb{R}$

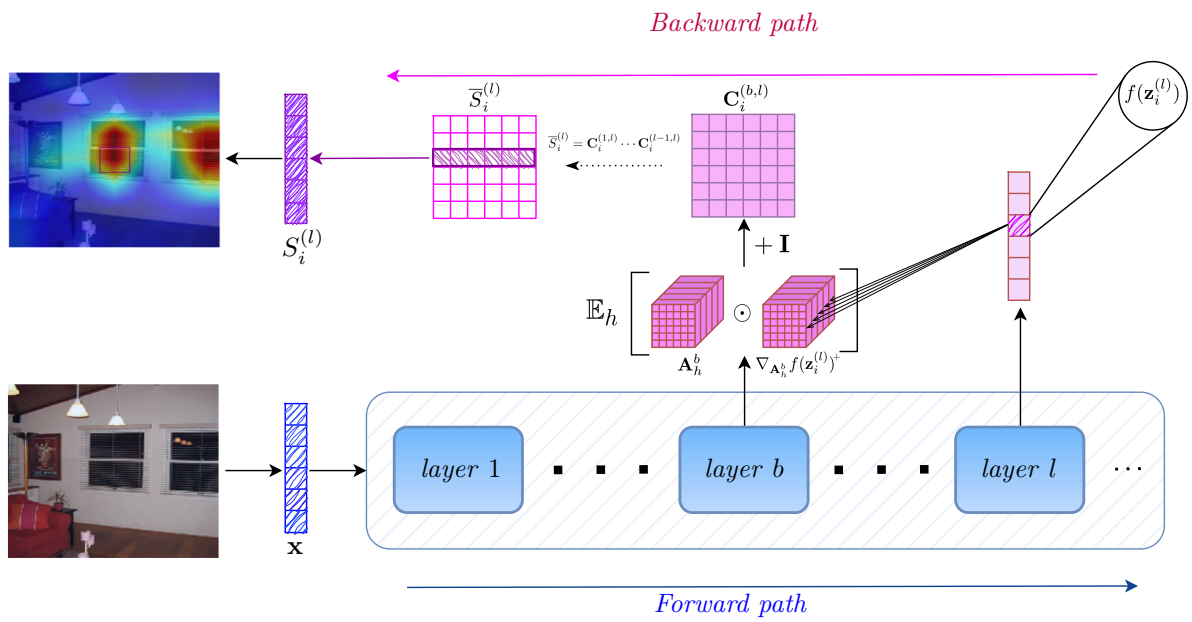


Figure 1: The overall architecture of the `ULTra` framework. The framework consists of a forward path, where the input data $\mathbf{x}$ is fed into the model, and a backward path, starting from the target layer l, where we compute the gradient of a scalar function of the i-th latent token, $f(\mathbf{z}_i^{(l)})$, with respect to the attention probability matrix of the middle layer b. Next, we compute the corresponding contribution map $\mathbf{C}_i^{(b,l)}$ for all middle layers. Finally, we construct the explanation map $\overline{S}_i^{(l)}$, select its i-th row, and transform it to the input size.

of the target token. Specifically, by *influence*, we mean the degree to which an input token affects $f(\mathbf{z}_i^{(l)})$, reflecting its importance in the representation.

Finally, we define the corresponding explanation map for the latent token $\mathbf{z}_i^{(l)}$, denoted as $S_i^{(l)} \in \mathbb{R}^n$, where each element represents the influence of an input token on $\mathbf{z}_i^{(l)}$. The explanation map is computed as:

$$\overline{S}_i^{(l)} = \mathbf{C}_i^{(1,l)} \cdot \mathbf{C}_i^{(2,l)} \cdots \mathbf{C}_i^{(l-1,l)}, \quad S_i^{(l)} = \overline{S}_i^{(l)}[i, 1:], \tag{5}$$

where . represents matrix multiplication, which aggregates contributions from all layers to the target latent token. Transformer skip connections cause most contributions to concentrate on $S_i^{(l)}[i-1]$, hindering token-level analysis. To mitigate this, we replace it with the maximum of the other elements, better capturing token contributions. Moreover, in some experiments, e.g., semantic segmentation, we reshape and upsample the explanation map using bilinear or cubic interpolation to match the input resolution, producing $\tilde{S}_i^{(l)}$. The overall framework of `ULTra` is shown in Figure 1.

Additionally, As we illustrated in section 5, in our experimetns we utilize three approach for desiging f in `ULTra` framework:

(i) $f_{\mathbf{s}}(\mathbf{z}_i^{(l)}) = \langle \mathbf{z}_i^{(l)}, \mathbf{1} \rangle$ In this formulation, $\mathbf{1}$ represents an all-ones vector, meaning that the function simply computes the sum of all elements in $\mathbf{z}_i^{(l)}$. This approach is denoted by ULTra$_{\mathcal{S}}$. The underlying intuition is that this approach treats all elements of the token equally.

(ii) $f_{\mathbf{e}}(\mathbf{z}_i^{(l)}) = \langle \mathbf{z}_i^{(l)}, \mathbf{z}_i^{(l)} \rangle$ This approach is based on the energy (or norm) of a given token, a concept that has been previously explored in the literature Darcet et al. (2024). By leveraging token energy, this method captures the magnitude of the token's latent representation. We refer to this variant as ULTra$_{\mathcal{E}}$.

(iii) $f_{\mathbf{w}}(\mathbf{z}_i^{(l)}) = \langle \mathbf{z}_i^{(l)}, \mathbf{w}_i \rangle$: Besides of previous approaches, we introduce a learnable vector $\mathbf{w}_i$ to project each token $\mathbf{z}_i^{(l)}$. Geometrically, the functionality of $f_{\mathbf{s}}$ in ULTra$_{\mathcal{S}}$ can be seen as mapping all tokens onto the

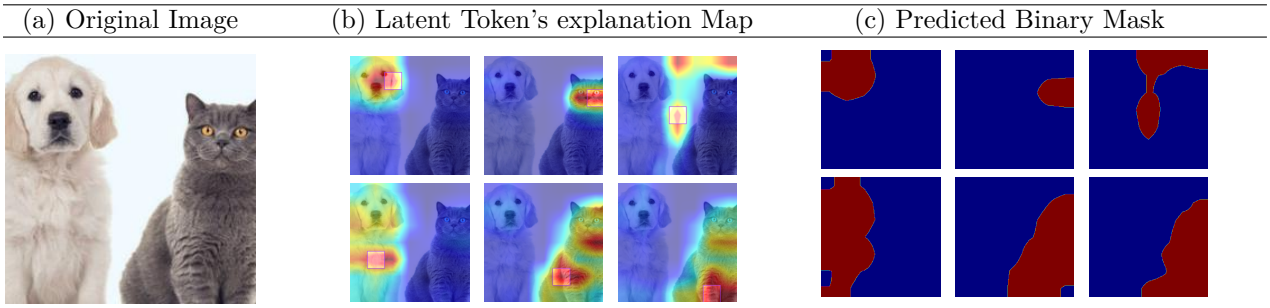


Figure 3: An example of token interpretation by our model and its predicted binary mask. (a) Original image. (b) Overlay of $\tilde{S}_i^{(13)}$ on the original image for different i, where the location of the i-th token is indicated by the purple square. (c) The binary mask $M_i^{(13)}$ for each corresponding explanation map in (b).

same projection basis within the token space. Intuitively, the goal of this approach is to enhance the overall performance of the `ULTra` framework by computing a more informative projection basis. We refer to this variant as ULTra$_{\mathcal{W}}$. While ULTra$_{\mathcal{S}}$ is training-free, ULTra$_{\mathcal{W}}$ requires light training to learn the optimal $\mathbf{w}_i$ vectors. In our experiments, we train these vectors using a limited number of batches. Notably, we employ this approach to further enhance segmentation performance. Furthermore, in Section 4.1, we propose a self-supervised approach for learning $\mathbf{w}_i$.

For both ULTra$_{\mathcal{S}}$ and ULTra$_{\mathcal{E}}$, our framework operates without requiring any additional training. In the case of semantic segmentation, experimental results demonstrate that these proposed approaches closely match, or even surpass, other baselines that rely on fine-tuning, highlighting their effectiveness and efficiency.

4 Tailoring ULTra for Different Applications

In this section, we aim to examine `ULTra`'s capability to adapt to various tasks involving semantic knowledge.

4.1 Unsupervised Semantic Segmentation

Explanation maps are defined for each latent token at a fixed layer, resulting in a total of as many maps as there are latent tokens. For segmentation, we hierarchically cluster these maps. In our experiments, we set a predefined number of clusters k for the algorithm. Additionally, in Appendix E, we conduct comprehensive experiments exploring the use of a threshold ζ to dynamically adjust the optimal cluster count k. Figure 12 also provides a schematic of how it is done. To ensure fair representation, we apply Min-Max scaling to prevent larger objects from dominating the clustering process.

After clustering, k distinct concepts are defined by aggregating explanation maps. The aggregated explanation map for cluster c is:

$$\tilde{S}_c^{(l)}[x,y] = \sum_{i \in \phi(c)} \tilde{S}_i^{(l)}[x,y], \tag{6}$$

where $\phi(c) = \{i : \text{Class}(\tilde{S}_i^{(l)}) = c\}$ represents the grouping of label assignments. Given a fixed threshold ζ, class labels are assigned to input pixels using the explanation map of the l-th layer as follows:

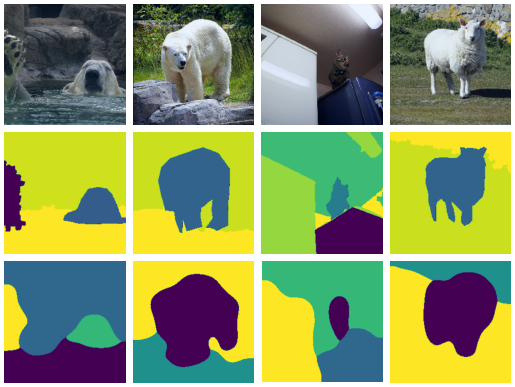


Figure 2: ULTra segmentation results on sample images. The top row displays the original images, the middle row shows true annotations, and the bottom row presents our model's predictions.

$$\text{Class}[x,y]^{(l)} = \underset{c \in \{1,\dots,k\}}{\text{argmax}}\ S_c^{(l)}[x,y]. \tag{7}$$

Some examples illustrating our segmentation method are presented in Figure 2.

ULTra$_{\mathcal{W}}$. As previously mentioned, to further enhance segmentation performance, we propose ULTra$_{\mathcal{W}}$, which utilizes a learnable transformation matrix to obtain a more informative projection basis for the token space. For simplicity of notation, we fix the layer ℓ and omit the layer superscript in the following derivations. Consider $z_{i,j} = \mathbf{e}_j^T \mathbf{z}_i$, where $\mathbf{e}_j$ is the j-th standard basis vector. The gradients can then be rewritten as:

$$\nabla_{\mathbf{A}_h^b} f_{\mathbf{s}}(\mathbf{z}_i) = \sum_j \nabla_{\mathbf{A}_h^b} z_{i,j}, \quad \nabla_{\mathbf{A}_h^b} f_{\mathbf{w}}(\mathbf{z}_i) = \sum_j \mathbf{w}_{i,j} \nabla_{\mathbf{A}_h^b} z_{i,j}. \tag{8}$$

In other words, the gradient of $f_{\mathbf{w}}(\mathbf{z}_i)$ is a weighted version of the gradient of $f_{\mathbf{s}}(\mathbf{z}_i)$. From this perspective, by manipulating the weighting vector $\mathbf{w}_i$, we can prioritize the more informative dimensions of the token space. As illustrated in section 5, this leads to improved performance of ULTra compared to variants that rely on a naive choice of f.

Here, a crucial question arises: *How do we characterize the amount of information in the token space to achieve a good projection basis for the token space?* In other words, we need to learn a matrix $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_n]^T$ to perform more effective segmentation. To this end, we introduce a perturbation on the input image and, by optimizing $\mathbf{W}$, aim to minimize the divergence of specific tokens from their unperturbed versions.

Let $\mathbf{x}$ be an input image and $\tilde{\mathbf{x}}$ its perturbed version corresponding to segmentation class c, which can be computed as:

$$\tilde{\mathbf{x}}^c = \mathbf{x} + \mathcal{P}_\phi(\mathbf{x}, c; \delta), \tag{9}$$

where $\mathcal{P}_\phi$[1] is a perturbation strategy that applies perturbations to the corresponding negative regions predicted for class $c \in \mathcal{C}$. Moreover, δ quantifies the amount of perturbation, and for $\delta = 0$, we have $\tilde{\mathbf{x}}^c = \mathbf{x}$.

Now, to optimize the parameters $\mathbf{W}$, we formulate the problem as:

$$\min_{\mathbf{W}} \mathbb{E}_{x \sim \mathcal{D}} \left[\sum_{c \in \mathcal{C}} \sum_{i \in \{j | \mathbf{x}_j = \mathbf{x}_j^c\}} d_{\mathbf{W}}(\mathbf{z}_i, \tilde{\mathbf{z}}_i) \right], \quad \text{s.t.} \quad \|\mathbf{w}_k\|_2 = 1, \ \forall k \in \{1, \cdots, n\}, \tag{10}$$

where $d_{\mathbf{W}}(\mathbf{z}_i, \tilde{\mathbf{z}}_i)$ denotes the divergence between the two vectors based on the projection basis $\mathbf{w}_i$, $\|\cdot\|_2$ is the eculidian norm, and $\mathbf{x}_i$ and $\mathbf{z}_i$ represent the i-th input token and latent token, respectively. In particular, we do not perturb the regions recognized by ULTra$_{\mathcal{W}}$ that contain localized information. This encourages the model to become more aware of self-attended regions, ultimately enhancing its decision-making in these critical areas.

In our experiments, we parameterize each projection vector as $\mathbf{w}_k = \boldsymbol{\theta}_k / \|\boldsymbol{\theta}_k\|_2$. Hence, based on the formulation in Equation 10, to optimize the parameters $\boldsymbol{\theta}_k$, we define the *Self-Consistency* loss as follows:

$$\mathcal{L}_{\text{sc}}(x, \boldsymbol{\Theta}) = \sum_{c \in \mathcal{C}} \sum_{i \in \{j | \mathbf{x}_j = \mathbf{x}_j^c\}} \left| \mathbf{w}_i^T (\mathbf{z}_i - \tilde{\mathbf{z}}_i) \right|^2, \tag{11}$$

where $\mathbf{w}_i = \boldsymbol{\theta}_i / \|\boldsymbol{\theta}_i\|_2$ and $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n]^T$. We use the term *"Self"* because we leverage the model's own segmentation results, obtained using ULTra$_{\mathcal{W}}$, to enhance its performance through a systematic weighting approach. Experimentally, we found that lightly optimizing $\boldsymbol{\Theta}$ on a limited number of samples improves the model's segmentation performance.

4.2 Latent Token Interpretability Assessment

While semantic segmentation serves as an interpretability evaluation approach for `ULTra`, in this section, we evaluate `ULTra` in additional introduced metrics. Specifically, we designed two tasks: (i) Object Selection and (ii) Perturbation Test. Additionally, in Section 5.3, we propose quantitative metrics to evaluate our

[1]The perturbation strategy is defined based on the model parameters ϕ since the model is involved in distinguishing the perturbation using its segmentation.

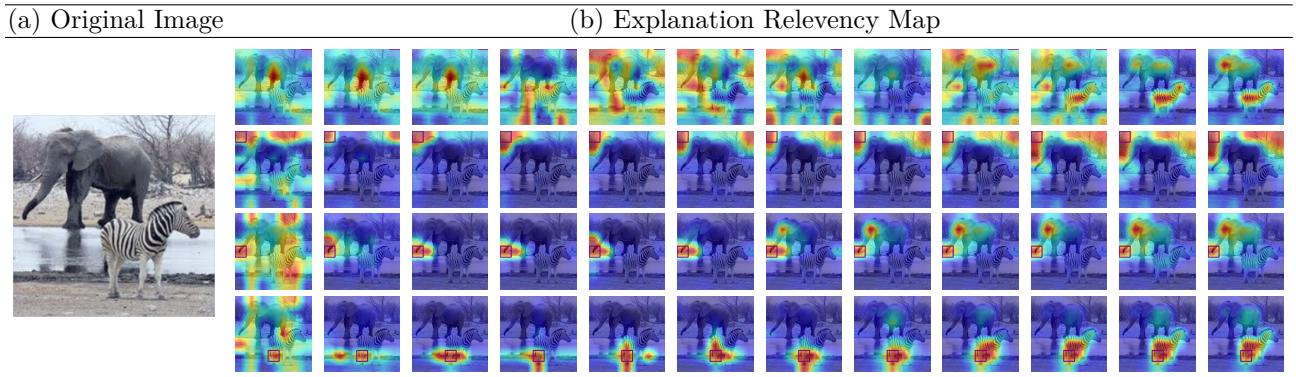


Figure 4: An example illustrating the model's decision-making process across layers. Moving from left to right corresponds to deeper layers in the network. The first row corresponds to the `CLS` token, while the second, third, and fourth rows represent three different tokens, highlighted by red squares.

method on these tasks to provide deeper insights into the interpretability of `ULTra` by identifying influential regions and behaviors within the model's decision-making process.

Perturbation Test. The perturbation test is a widely used metric for evaluating explainability methods Chefer et al. (2021b); Wu et al. (2024). This approach involves perturbing parts of an image based on the explanation map generated by the method and measuring the resulting change in the predicted class probability. Inspired by this technique, instead of using class probability, we leverage the change in the embedding space to design a perturbation-based validation test. This allows us to assess the reliability of the explanation maps generated for the latent tokens.

Object Selection. In this task, we convert the upsampled explanation map $\tilde{S}_i^{(l)}$ into a binary segmentation mask using a threshold τ, where the binary mask $M_i^{(l)}$ is defined as:

$$M_i^{(l)}[x,y] = \begin{cases} 0, & \text{if } \tilde{S}_i^{(l)}[x,y] < \tau, \\ 1, & \text{otherwise.} \end{cases} \tag{12}$$

Here, $\tilde{S}_i^{(l)}[x,y]$ represents the relevance value at position $[x,y]$ in $\tilde{S}_i^{(l)}$, with τ as the threshold. When $M_i^{(l)}[x,y] = 1$, it indicates that the position $[x,y]$ belongs to the object region.

Our findings indicate that as tokens propagate through the network, they refine their object representation while retaining the semantic meaning of their associated image patches performing *Object Selection*. Figure 4 visually illustrates this process. For a given patch token $\mathbf{x}_i$, the object it most strongly represents is denoted as class k_i. The latent token $\mathbf{z}_i^{(l)}$ generates an explanation map that highlights areas with higher values associated with class k_i in the whole image, including $\mathbf{x}_i$. After applying a threshold, this map becomes a binary segmentation mask expected to exhibit a high Intersection over Union (IoU) with the corresponding class k_i region in the image. An illustrative example is shown in Figure 3.

4.3 Interpreting LLMs in Text Summarization

In this section, we examine how our interpretability framework can be applied to text summarization tasks, taking steps toward uncovering the underlying intent of LLMs. In Section 5.4, we qualitatively evaluate ULTra$_{\mathcal{S}}$ using two samples by visualizing the regions of the input context that an LLM prioritizes while interpreting a given TL;DR summary. This analysis reveals key input regions shaping the model's decisions, aiding in concise and relevant summary generation.

For this task, we concatenate the context x and the summary y with a separator token. After feeding this input into the model, we compute the relevance scores of the TL;DR tokens with respect to the context tokens. We then average these scores for each token in x to obtain a scalar value, referred to as the Token

Contribution Score, $\lambda_i^{(l)} \in \mathbb{R}^+$, which highlights the contribution of each context token in interpreting the summary y within the given context. Accordingly, $\lambda_i^{(l)}$ is computed as:

$$\lambda_i^{(l)} = \frac{1}{|y|} \sum_{j=1}^{|y|} S_{j+|x|}^{(l)}[i], \quad \forall i \in \{1, \cdots, |x|\}, \tag{13}$$

where $|\cdot|$ denotes the number of tokens in the text.

5 Experiments & Results

5.1 Experimental Setup

Datasets. In our experiments, we evaluate model performance on several semantic segmentation benchmarks, focusing on vision-related tasks. We conducted experiments on four datasets: COCO-Stuff 27 Caesar et al. (2018), PASCAL VOC 2012 Everingham & Winn (2011), Potsdam-3 ISPRS (2018), and Cityscapes Cordts et al. (2016).

For our qualitative analysis of LLM interpretation in the task of text summarization, as described in Section 4.3, we utilized the TL;DR dataset (Stiennon et al., 2022). The TL;DR dataset contains summary comparisons with human feedback collected by OpenAI.

Models. For all experiments in the vision tasks, we used different pretrained versions of CLIP's image encoder (Radford et al., 2021) as well as DINO ViT-S/16 and ViT-B/16 (Caron et al., 2021). For interpreting text summarization, as described in Section 4.3, we used the Llama-2-7B language model (Touvron et al., 2023). All experiments were run on 4 NVIDIA A100-80GB GPUs.

Further details of the datasets and models used are provided in Appendix C. red

5.2 Semantic Segmentation

We benchmarked the segmentation performance of our method against several SOTA approaches in the literature on unsupervised segmentation. To train the matrix W for ULTra$_{\mathcal{W}}$, we utilized 256 samples.

Method	Model	Training	U. ACC	U. mIoU
IIC	R18+FPN	✓	21.8	6.7
PiCIE	R18+FPN	✓	48.1	13.8
DINO	ViT-S/8	✓	28.7	11.3
	ViT-S/16	✓	22.0	8.0
	ViT-B/8	✓	30.5	9.6
ACSeg	ViT-S/16	✓	-	16.4
TransFGU	ViT-S/8	✓	52.7	17.5
STEGO	ViT-S/8	✓	48.3	24.5
	ViT-S/16	✓	52.5	23.7
	ViT-B/8	✓	56.9	28.2
STEGO +HP	ViT-S/8	✓	57.2	24.6
	ViT-S/16	✓	54.5	24.3
DepthG	ViT-S/8	✓	56.3	25.6
	ViT-B/8	✓	58.6	29.0
U2Seg	R50	✓	63.9	30.2
ULTra$_{\mathcal{W}}^{\text{CLIP}}$	ViT-B/32	✓	60.6	34.1
	ViT-B/16	✓	63.8	34.0
	ViT-L/14	✓	**67.9**	**38.2**
ULTra$_{\mathcal{S}}^{\text{CLIP}}$	ViT-B/32	✗	60.8	34.6
	ViT-B/16	✗	63.0	33.2
	ViT-L/14	✗	66.5	37.5
ULTra$_{\mathcal{E}}^{\text{CLIP}}$	ViT-B/32	✗	59.5	32.6
	ViT-B/16	✗	53.6	26.6
	ViT-L/14	✗	59.0	31.7
ULTra$_{\mathcal{W}}^{\text{DINO}}$	ViT-S/16	✓	67.2	34.4
	ViT-B/16	✓	67.4	37.7
ULTra$_{\mathcal{S}}^{\text{DINO}}$	ViT-S/16	✗	66.4	33.3
	ViT-B/16	✗	67.3	35.6
ULTra$_{\mathcal{E}}^{\text{DINO}}$	ViT-S/16	✗	63.4	31.6
	ViT-B/16	✗	63.0	31.3

Table 1: Comparison of unsupervised segmentation methods on the COCO-Stuff dataset.

To evaluate the effectiveness of our approach, we use the Unsupervised mean Intersection over Union (U. mIoU) and Unsupervised Pixel Accuracy (U. ACC) metrics. The experimental results on the COCO-Stuff, PASCAL VOC, Potsdam, and Cityscapes datasets are reported in Tables 1, 2, 3, and 4, respectively. In these tables, the Training column indicates whether any additional training is required.

Method	Model	Training	U. mIoU
IIC	R18+FPN	✓	6.4
PiCIE	R18+FPN	✓	12.3
STEGO	ViT-B/8	✓	21.0
STEGO +HP	ViT-S/8	✓	18.4
	ViT-B/8	✓	18.4
DepthG	ViT-B/8	✓	23.1
$\text{ULTra}_{\mathcal{W}}^{\text{CLIP}}$	ViT-B/32	✓	17.6
	ViT-B/16	✓	24.8
	ViT-L/14	✓	25.1
$\text{ULTra}_{\mathcal{S}}^{\text{CLIP}}$	ViT-B/32	✗	17.1
	ViT-B/16	✗	24.2
	ViT-L/14	✗	24.2
$\text{ULTra}_{\mathcal{E}}^{\text{CLIP}}$	ViT-B/32	✗	20.4
	ViT-B/16	✗	20.9
	ViT-L/14	✗	23.0
$\text{ULTra}_{\mathcal{W}}^{\text{Dino}}$	ViT-S/16	✓	25.8
	ViT-B/16	✓	**26.5**
$\text{ULTra}_{\mathcal{S}}^{\text{Dino}}$	ViT-S/16	✗	24.2
	ViT-B/16	✗	25.7
$\text{ULTra}_{\mathcal{E}}^{\text{Dino}}$	ViT-S/16	✗	22.9
	ViT-B/16	✗	23.0

Table 2: Comparison of different unsupervised segmentation methods on the Cityscapes dataset.

Method	Model	Training	U. ACC
IIC	R18+FPN	✓	65.1
DINO	ViT-S/8	✓	71.3
STEGO	ViT-S/8	✓	77.0
DepthG	ViT-S/8	✓	80.4
$\text{ULTra}_{\mathcal{W}}^{\text{CLIP}}$	ViT-B/32	✓	78.7
	ViT-B/16	✓	80.9
	ViT-L/14	✓	82.4
$\text{ULTra}_{\mathcal{S}}^{\text{CLIP}}$	ViT-B/32	✗	78.3
	ViT-B/16	✗	80.9
	ViT-L/14	✗	**82.8**
$\text{ULTra}_{\mathcal{E}}^{\text{CLIP}}$	ViT-B/32	✗	78.1
	ViT-B/16	✗	70.4
	ViT-L/14	✗	75.5
$\text{ULTra}_{\mathcal{W}}^{\text{Dino}}$	ViT-S/16	✓	79.0
	ViT-B/16	✓	80.7
$\text{ULTra}_{\mathcal{S}}^{\text{Dino}}$	ViT-S/16	✗	77.4
	ViT-B/16	✗	80.8
$\text{ULTra}_{\mathcal{E}}^{\text{Dino}}$	ViT-S/16	✗	75.7
	ViT-B/16	✗	76.8

Table 3: Comparison of different unsupervised segmentation methods on the Potsdam dataset.

Among the proposed variants of ULTra, $\text{ULTra}_{\mathcal{W}}$ achieves the highest performance, requiring only a small number of training samples. Notably, even when no training data is available, $\text{ULTra}_{\mathcal{S}}$ still achieves state-of-the-art results on several benchmarks.

for some models, such as ViT-L/14, no existing baseline is available for direct comparison, highlighting the versatility of `ULTra` across different architectures. Furthermore, we conducted an ablation study related to the model depth in Appendix A.

5.3 Interpretability Evaluations

Perturbation Test. To assess the reliability of the explanation maps, we conduct a perturbation test by selectively altering image regions based on the explanation map $S_i^{(l)}$ for each token i. The perturbation is applied at the patch level while ensuring the total perturbed area remains consistent across all cases. We consider two types of perturbations:

(i) Positive Perturbation: Removing highly relevant regions, which should significantly affect the token's representation.

Method	Model	Training	U. mIoU
IIC	R18+FPN	✓	9.8
MaskContrast	R50	✓	35.0
Leopart	ViT-S/16	✓	41.7
TransFGU	ViT-S/8	✓	37.2
MaskDistill	ViT-S/16 + R50	✓	42.0
ACSeg	ViT-S/16	✓	47.1
$\text{ULTra}_{\mathcal{W}}^{\text{CLIP}}$	ViT-B/32	✓	**51.2**
	ViT-B/16	✓	50.9
	ViT-L/14	✓	49.1
$\text{ULTra}_{\mathcal{S}}^{\text{CLIP}}$	ViT-B/32	✗	49.2
	ViT-B/16	✗	48.3
	ViT-L/14	✗	48.7
$\text{ULTra}_{\mathcal{E}}^{\text{CLIP}}$	ViT-B/32	✗	50.0
	ViT-B/16	✗	40.0
	ViT-L/14	✗	45.2
$\text{ULTra}_{\mathcal{W}}^{\text{DINO}}$	ViT-S/16	✓	48.9
	ViT-B/16	✓	50.5
$\text{ULTra}_{\mathcal{S}}^{\text{DINO}}$	ViT-S/16	✗	47.9
	ViT-B/16	✗	50.0
$\text{ULTra}_{\mathcal{E}}^{\text{DINO}}$	ViT-S/16	✗	46.9
	ViT-B/16	✗	50.0

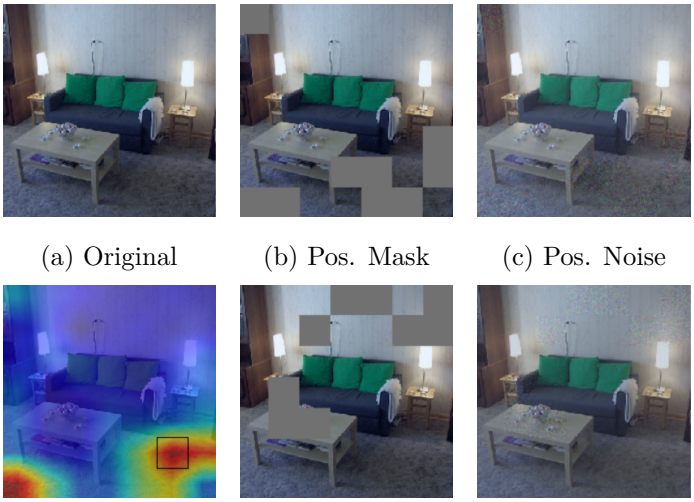
Table 4: Comparison on the PASCAL VOC 2012 dataset.

Model	Perturb.	VOC		Potsdam		COCO		Cityscapes	
		Neg. ↓	Pos. ↑	Neg. ↓	Pos. ↑	Neg. ↓	Pos. ↑	Neg. ↓	Pos. ↑
ViT-B/32	Mask	13.16	15.51	12.35	14.02	12.62	15.21	9.95	14.42
	Noise	6.97	10.23	9.39	11.44	6.83	10.26	5.99	11.13
ViT-B/16	Mask	14.68	17.5	13.91	15.51	14.16	17.39	11.33	16.27
	Noise	9.87	14.03	13.5	14.45	10.14	14.12	9.46	14.22
ViT-L/14	Mask	6.66	9.43	6.61	8.99	6.38	9.31	5.36	8.86
	Noise	4.45	7.88	4.95	7.96	4.27	7.74	3.89	7.77

Table 5: Average token vector differences for ViT models under mask and noise perturbation tests across multiple datasets, highlighting the impact of positive and negative perturbations based on the relevancy map on token representations.

(ii) Negative Perturbation: Removing less relevant regions, which should have minimal impact.

Two types of modifications are introduced. In the masking perturbation, selected patches are replaced with zeros, effectively removing the corresponding visual information. This is applied to both highly relevant regions (positive masking) and less relevant areas (negative masking). In contrast, the noise perturbation introduces Gaussian noise to the same sets of patches, adding controlled randomness to test the robustness of token representations.



(a) Original (b) Pos. Mask (c) Pos. Noise

(d) Explain. Map (e) Neg. Mask (f) Neg. Noise

Figure 5: Effect of perturbations on a sample image from the PASCAL VOC dataset, where the model used is CLIP ViT-B/32.

The noise follows a standard normal distribution with a standard deviation of 0.3. Figure 5 provides a detailed visualization of the perturbation test conducted on a sample image from the PASCAL VOC dataset using the CLIP ViT-B/32 model. It illustrates the model's explanation map (d), and compares the effects of both positive (b, c) and negative (e, f) perturbations applied through masking and Gaussian noise, demonstrating how altering semantically relevant regions leads to more significant changes in the model's internal representations.

To ensure fairness, the initial token's patch remains unchanged in both perturbations, as its information is directly propagated to the target token through skip connections. Given a perturbed representation $\tilde{z}_i^{(l)}$, we measure the deviation from the original token representation $z_i^{(l)}$ using the Euclidean distance, then we compute the average deviation over the entire dataset by selecting k random tokens per image and aggregating the distances:

$$\bar{d}_{\text{Euc}} = \frac{1}{M} \sum_{j=1}^{M} \left(\frac{1}{k} \sum_{i=1}^{k} \| z_{i,j}^{(l)} - \tilde{z}_{i,j}^{(l)} \|_2 \right), \tag{14}$$

where M is the number of images, and k tokens are randomly selected from each. For our results, we set $k = 10$. The perturbation test results are presented in Table 5

Positive perturbations result in greater deviations compared to negative ones, confirming that the explanation maps effectively highlight influential regions. The consistency of these results across multiple datasets further validates the reliability of our interpretability framework.

Object Selection. To quantify alignment, we compute the IoU by converting the explanation map $S_i^{(l)}$ into a binary mask $M_i^{(l)}$ and comparing it with the ground-truth mask. We propose the *Initial Token IoU (ITIoU)* metric, which measures how well the explanation maps of input tokens align with their respective class masks. The $ITIoU$ is calculated as:

$$ITIoU^{(l)}(X) = \frac{1}{C}\sum_{i=1}^{C}\frac{1}{|\mathcal{T}_i|}\sum_{\mathbf{x}_j\in\mathcal{T}_i}\text{IoU}(M_j^{(l)}, G_i), \tag{15}$$

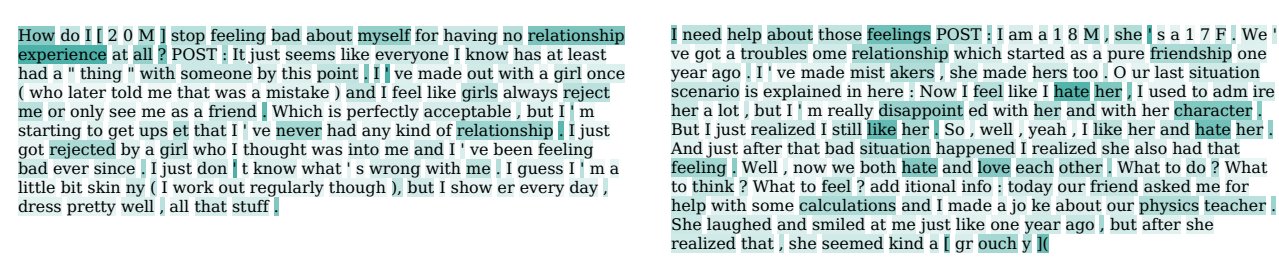
where C denotes the number of classes, $\mathcal{T}_i$ represents the set of tokens associated with class i, $M_j^{(l)}$ is the binary segmentation mask for token $\mathbf{x}_j$ within class i, and G_i is the ground-truth mask for class i in image $\mathbf{x}$. The inner sum averages the IoU for tokens in $\mathcal{T}_i$ for each class, and the outer sum then averages across all classes. Using a threshold of 0.2, our $ITIoU$ metric achieves an average score of 37.84% % on the COCO-Stuff validation dataset and 39.51% % on the PASCAL VOC dataset. A more detailed analysis of $ITIoU$ is provided in Appendix D.

5.4 Interpretable Text Summarization

In this experiment, we used a Supervised Fine-Tuned (SFT) version of Llama-2-7B trained on the UltraFeedback Binarized (UFB) dataset (Cui et al., 2024). Additionally, we aligned the model to the text summarization task on the TL;DR dataset (Stiennon et al., 2022) using the Direct Preference Optimization (DPO) method (Rafailov et al., 2024) for 1,000 iterations, with a learning rate of 5×10^{-6} and $\beta = 0.5$. To validate our framework, we selected the *preferred* response (TL;DR) of each sample in the dataset, denoted by y, and used it as the summary of the context x. the result can be seen in figure 6

In example (a): semantically significant words such as 'relationship', 'experience', 'rejection', and 'never' are prominently highlighted, reflecting the model's interpretation of the person's struggles with relationships and feelings of rejection. Additionally, the highlighting of the question at the beginning of the context 'How do I stop feeling bad...' suggests the model recognizes the presence of uncertainty and a request for guidance, which is encapsulated in the summary as 'I don't know.'

In example (b): $\lambda_i^{(l)}$ scores reveals the model's focus on words such as 'feelings', 'hate', 'disappoint', 'love', and 'like', which correspond to the person's mixed emotions toward their girlfriend, as described in the summary. The apparent contradiction between 'love' and 'trashness' in the summary seems to be derived from these highlighted terms, suggesting the model understands the conflicting emotions present in the text. Furthermore, the focus on 'character' reflects the summary's judgmental tone, implying the model links it to personality assessment.



(a) TL;DR: I've had very bad luck with girls my whole life and I don't know how to get my confidence up.

(b) TL;DR: I still like her but my rational side says "no, she is a trash person".

Figure 6: Visualization of Token Contribution Scores ($\lambda_i^{(l)}$) highlighting the relevance of context tokens in interpreting the summary. Each token is colored proportionally to its $\lambda_i^{(l)}$ value. These visualizations demonstrate the model's ability to identify key semantic elements in the context for generating relevant summaries.

This token-level analysis visually illustrates the model's context processing, supporting interpretable fine-tuning and alignment techniques such as RLHF (Christiano et al., 2023; Stiennon et al., 2022; Ouyang et al., 2022) and direct preference optimization (Rafailov et al., 2024; Azar et al., 2023; Ethayarajh et al., 2024), where understanding model behavior and intent is critical. These language-based experiments are not intended as a benchmark for interpretable summarization, especially given the absence of standardized evaluation metrics, but rather to demonstrate the versatility of ULTra across modalities. This experiments also highlights an important direction for future research: applying post-hoc interpretability techniques to better understand and align large language models within frameworks such as RLHF and preference-based training.

6 Concluding Remarks

Summary. We present a framework for interpreting latent tokens in Transformers. We investigate several design choices for `ULTra`, and our method achieves state-of-the-art performance in unsupervised semantic segmentation across multiple datasets and settings without any training. Additionally, we validate our approach through perturbation tests and object selection, demonstrating its broader applicability. This framework offers valuable insights into the decision-making processes and behavior of Transformer models at the layer level.

Future Research. As a contribution to latent token interpretability, our framework extends existing methods. Future directions include refining the method specifically for latent token analysis and addressing its computational complexity during inference (further discussion is provided in Appendix B.2).

References

Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*, 2020.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences, 2023. URL `https://arxiv.org/abs/2310.12036`.

Daniel Bolya, Jian Wang, Ambrish Pujari, Ross Wightman, and Alexander Kirillov. Token merging: Your vit but faster. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3838–3848, 2023.

Walid Bousselham, Angie Boggust, Sofian Chaybouti, Hendrik Strobelt, and Hilde Kuehne. Legrad: An explainability method for vision transformers via feature formation sensitivity. *arXiv preprint arXiv:2404.03214*, 2024.

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Computer vision and pattern recognition (CVPR), 2018 IEEE conference on*. IEEE, 2018.

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.

Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 387–396. IEEE, October 2021a. doi: 10.1109/iccv48922.2021.00045. URL `http://dx.doi.org/10.1109/ICCV48922.2021.00045`.

Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 782–791, 2021b.

Haozhe Chen, Junfeng Yang, Carl Vondrick, and Chengzhi Mao. Invite: Interpret and control vision-language models with text explanations. In *The Twelfth International Conference on Learning Representations*, 2024.

Tao Chen, Yazhou Yao, Lei Zhang, Qiong Wang, Guo-Sen Xie, and Fumin Shen. Saliency guided inter- and intra-class relation constraints for weakly supervised semantic segmentation. *IEEE Transactions on Multimedia*, 25:1727–1737, 2023. ISSN 1941-0077. doi: 10.1109/tmm.2022.3157481. URL `http://dx.doi.org/10.1109/TMM.2022.3157481`.

Jang Hyun Cho, Utkarsh Mall, Kavita Bala, and Bharath Hariharan. Picie: Unsupervised semantic segmentation using invariance and equivariance in clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16794–16804, 2021.

Jaegul Choe and Hyunjung Shim. Attention-based dropout layer for weakly supervised object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2219–2228, 2019.

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences, 2023. URL `https://arxiv.org/abs/1706.03741`.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with scaled ai feedback, 2024. URL `https://arxiv.org/abs/2310.01377`.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=2dnO3LLiJ1`.

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

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization, 2024. URL `https://arxiv.org/abs/2402.01306`.

Mark Everingham and John Winn. The pascal visual object classes challenge 2011 (voc2011) development kit. *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep*, 8, 2011.

Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T Freeman. Unsupervised semantic segmentation by distilling feature correspondences. *arXiv preprint arXiv:2203.08414*, 2022.

ISPRS. 2d semantic labeling contest - potsdam. `https://www.isprs.org/education/benchmarks/UrbanSemLab/2d-sem-label-potsdam.aspx`, 2018.

Sarthak Jain and Byron C Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019.

Xu Ji, Joao F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9865–9874, 2019.

Tsung-Wei Ke, Jyh-Jing Hwang, Yunhui Guo, Xudong Wang, and Stella X Yu. Unsupervised hierarchical semantic segmentation with multiview cosegmentation and clustering transformers. *arXiv preprint arXiv:2204.11432*, 2022.

Mengcheng Lan, Xinjiang Wang, Yiping Ke, Jiaxing Xu, Litong Feng, and Wayne Zhang. Smooseg: smoothness prior for unsupervised semantic segmentation. *Advances in Neural Information Processing Systems*, 36:11353–11373, 2023.

Seungju Lee, Kyumin Cho, Eunji Kwon, Sejin Park, Seojeong Kim, and Seokhyeong Kang. Vit-togo: Vision transformer accelerator with grouped token pruning. In *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1–6. IEEE, March 2024. doi: 10.23919/DATE58400.2024.10546804. URL http://dx.doi.org/10.23919/DATE58400.2024.10546804.

Kehan Li, Zhennan Wang, Zesen Cheng, Runyi Yu, Yian Zhao, Guoli Song, Chang Liu, Li Yuan, and Jie Chen. Acseg: Adaptive conceptualization for unsupervised semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7162–7172, 2023.

Youwei Liang, Chongjian GE, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. EVit: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=BjyvwnXXVn_.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

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, 2021.

Scott Lundberg. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

Dantong Niu, Xudong Wang, Xinyang Han, Long Lian, Roei Herzig, and Trevor Darrell. Unsupervised universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22744–22754, 2024.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Bingyi Pan, Dejia Zeng, Zhengping Shou, Zhiyu Yan, Lin Zhang, Gang Yuan, Houqiang Li, and Shuicheng Yan. Ia-red^2: Interpretability-aware redundancy reduction for vision transformers. In *Advances in Neural Information Processing Systems*, pp. 13845–13856, 2021.

Yao Qiang, Chengyin Li, Prashant Khanduri, and Dongxiao Zhu. Interpretability-aware vision transformer. *arXiv preprint arXiv:2309.08035*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL https://arxiv.org/abs/2305.18290.

Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, and Ziyu Yao. A practical review of mechanistic interpretability for transformer-based language models. *arXiv preprint arXiv:2407.02646*, 2024.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Leon Sick, Dominik Engel, Pedro Hermosilla, and Timo Ropinski. Unsupervised semantic segmentation through depth-guided feature correlation and sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3637–3646, 2024.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2014.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback, 2022. URL `https://arxiv.org/abs/2009.01325`.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open models based on gemini research and technology, 2024. URL `https://arxiv.org/abs/2403.08295`.

Hans Thisanke, Chamli Deshan, Kavindu Chamith, Sachith Seneviratne, Rajith Vidanaarachchi, and Damayanthi Herath. Semantic segmentation using vision transformers: A survey. *Engineering Applications of Artificial Intelligence*, 126:106669, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL `https://arxiv.org/abs/2307.09288`.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Unsupervised semantic segmentation by contrasting object mask proposals. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10052–10062, 2021.

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 (NeurIPS)*, 2017.

Jesse Vig and Yonatan Belinkov. Analyzing the structure of attention in a transformer language model. *arXiv preprint arXiv:1906.04284*, 2019.

Junyi Wu, Bin Duan, Weitai Kang, Hao Tang, and Yan Yan. Token Transformation Matters: Towards Faithful Post-Hoc Explanation for Vision Transformer . In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA, June 2024. IEEE Computer Society.

Jingkang Xu, Yifan Jiang, Zhenda Wang, Vijay Vasudevan, Xiaolong Liu, Tsung-Yi Lin, Song Bai, Luc Van Gool, and Qi Dai. Groupvit: Semantic segmentation emerges from text supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18134–18144, 2022.

Guangwei Yin, Xiaopeng Zhang, Lihui Liu, Jie Shao, Zhengming Chen, and Jianbo Wang. Dual-attention guided dropblock module for weakly supervised object localization. *arXiv preprint arXiv:2003.04719*, 2020.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *arXiv preprint arXiv:1311.2901*, 2014.

Adrian Ziegler and Yuki M Asano. Self-supervised learning of object parts for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14502–14511, 2022.

Appendix

A Effect of Layer Depth in ViT Token Understanding

In this section, we analyze the impact of depth on our model's interpretability and segmentation performance, providing insights into the contribution of each layer. for samller models such as ViT-B/32 in Figure 8, deeper layers generally carry more semantic significance. However, the contribution diminishes in the final layers, suggesting that a depth of around 13 layers might be more than sufficient for the ViT to effectively comprehend image content. This finding implies that even fewer layers might achieve comparable results, potentially reducing computational costs without compromising performance.

We observe an intriguing behavior in the initial layers, where performance initially declines before improving. This phenomenon is also visually evident in Figure 4, where the attention maps in the first layer appear to focus on the entire image. This suggests that, initially, the token examines the image as a whole before selectively gathering information from tokens with similar characteristics.

In the CLIP ViT-L/14 model, deeper layers capture more refined and detailed feature representations. Unlike shallower models, where segmentation performance stabilizes early, ViT-L/14 benefits from its depth by gradually extracting richer hierarchical features. As shown in Figure 7, accuracy and Mean IoU improve as layers deepen, with segmentation performance peaking around the mid-depth layers. However, in deeper layers (beyond layer 20), segmentation quality slightly declines. This suggests that while the model gains a better understanding of high-level semantics, it loses some spatial precision. This trade-off occurs because later layers prioritize semantic abstraction over detailed spatial structures.

While deeper models like ViT-L/14 offer improved feature extraction, more layers do not always lead to better segmentation. Instead, an optimal balance between depth and spatial representation is necessary for effective segmentation.

B Limitations & Discussions

B.1 Datasets

A significant challenge in deep learning is providing labeled datasets, which are essential for the success of deep learning methods. However, labeling can be ambiguous, as objects or attributes may be labeled separately

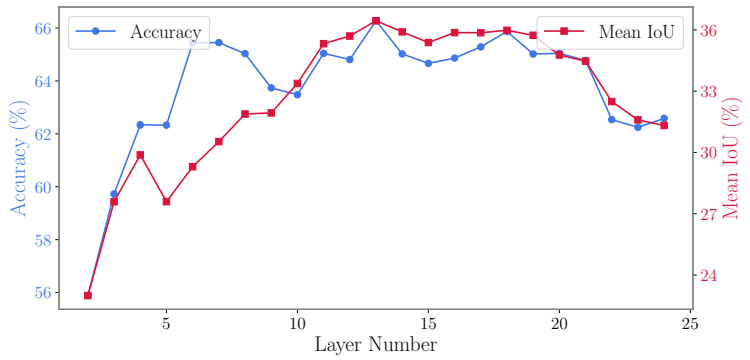
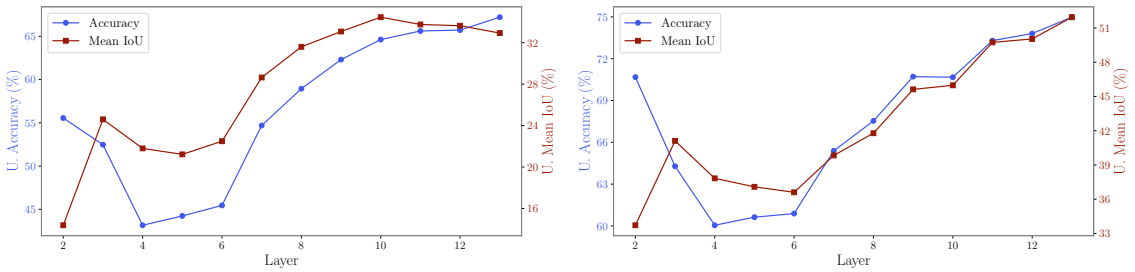


Figure 7: The segmentation performance of the CLIP ViT-L/14 model in an unsupervised setting, measured by accuracy and IoU, changes across different layers. As layer depth increases, both metrics improve at first, reaching their highest point in the mid-depth layers. However, performance slightly declines in deeper layers. This suggests that while deeper layers capture richer semantic details, they also introduce complexity that does not always improve segmentation. Additionally, the limitations of labeled datasets, especially the ambiguity in object definitions, further restrict the model's ability to achieve better segmentation, despite its strong interpretability.



(a) COCO-Stuff (b) PASCAL VOC

Figure 8: Ablation study on two evaluation metrics across layers. These plots demonstrate a progressive improvement in semantic segmentation performance in the deeper layers of the transformer model. This enhancement is attributed to latent tokens capturing more meaningful segment structures, resulting in increasingly accurate and refined semantic representations.

or combined as a single entity. Our method demonstrates how ViT interprets images in a zero-shot setting, capturing fine-grained or general representations. Notably, our method's predictions often exhibit logical consistency that surpasses the ground truth, effectively identifying relationships between objects. However, since our approach does not involve supervised training, it cannot adapt itself to detect only the specific objects labeled in the dataset. As a result, while the logical quality of the predictions is high, the numerical metrics may decline due to mismatches with the dataset's ground truth labels. Figure 9 illustrates this phenomenon with examples, showing instances where our method successfully captures unlabeled objects that are omitted in the dataset annotations.

B.2 Method

In this work, we aimed to uncover semantic information embedded within the latent tokens of transformers. A key challenge in interpretability is the evaluation process. In traditional model interpretation, where only class logits are available as the primary source of information, various evaluation approaches have been proposed. However, in our case, many of these metrics are not directly applicable due to the fundamental differences in our methodology.

To address this, we evaluated our approach in the context of unsupervised semantic segmentation. Although our baseline methods were originally designed for this task, we achieved superior results by aggregating our

Figure 9: This figure illustrates challenging samples of datasets COCO-Stuff 27, Potsdam-3, and PASCAL VOC that contribute to low performance due to poor dataset labeling quality. The first row displays the original images, the second row shows the ground truth labels and the third row presents our predicted segmentation.

explanation maps. This outcome highlights the potential for future research in this direction. Specifically, improved aggregation methods—such as clustering techniques—could enhance performance, particularly when dealing with smaller patch sizes. Notably, in the case of a patch size of 8, clustering algorithms failed due to the high number of tokens, making effective aggregation more challenging.

Moreover, despite being a zero-shot method, our approach incurs a significant computational cost when applied to semantic segmentation. This is primarily due to the necessity of computing multiple gradients for generating explanation maps, a process that must be performed for each token. As a result, inference time is considerably high. Future work could explore more efficient strategies to mitigate this issue, such as leveraging approximation techniques or optimizing the gradient computation process.

C Datasets & Models

C.1 Datasets

We utilize a combination of datasets to provide a diverse testing ground for evaluating our method across both standard and challenging perspectives in semantic segmentation and interpretability evaluations.

- **COCO-Stuff 27** Lin et al. (2014): A subset of the COCO dataset, featuring complex real-world scenes with pixel-level annotations across various object categories.
- **PASCAL VOC 2012** Everingham & Winn (2011): A widely used benchmark containing pixel-level annotations for foreground objects in structured scenes.
- **Potsdam-3**: A high-resolution aerial-view dataset capturing urban landscapes, including buildings, roads, and vegetation, presenting additional challenges due to its large-scale top-down perspective.
- **Cityscapes** Cordts et al. (2016): An urban street scene dataset with fine-grained pixel-level annotations, enabling the evaluation of segmentation performance in structured environments.

These datasets allow for a comprehensive evaluation of our approach across different environments, ensuring robustness across diverse segmentation challenges.

C.2 Models

We employ various **Vision Transformers (ViTs)** pre-trained on large-scale datasets. These models process images as non-overlapping patches and employ self-attention mechanisms across multiple layers to capture long-range dependencies.

Transformer Architectures

CLIP ViT Radford et al. (2021): A vision transformer trained using contrastive learning on 400 million image-text pairs. It encodes images into a shared embedding space with text prompts. CLIP variants include:

- **ViT-B/16**: Consists of 12 transformer layers, a hidden size of 768, and processes images with 16×16 patch resolution.
- **ViT-B/32**: Similar to ViT-B/16 but with a 32×32 patch resolution, reducing computational cost at the expense of finer details.
- **ViT-L/14**: A larger model with 24 transformer layers, a hidden size of 1024, and a 14×14 patch resolution, providing enhanced feature extraction.

DINO ViT Caron et al. (2021): A self-supervised vision transformer trained using knowledge distillation without labeled data. It learns image representations by maximizing similarity between different augmented views. Evaluated variants:

- **ViT-S/16**: A smaller model with 12 transformer layers, a hidden size of 384, and a patch size of 16×16.
- **ViT-B/16**: A larger model with 12 layers, a hidden size of 768, and a patch size of 16×16, providing stronger feature representation.

D ITIoU Analysis

This section evaluates the effectiveness of $ITIoU$ in assessing the performance of our object selection process. As expected, the final layers exhibit superior performance compared to the earlier layers, consistent with the results illustrated in Figure 11. This improvement highlights the increasing relevance of features in deeper layers for accurate object selection. Additionally, we observe the existence of an optimal threshold, τ, which significantly influences the segmentation performance. This phenomenon is depicted in Figure 10, where performance trends are analyzed across different threshold values.

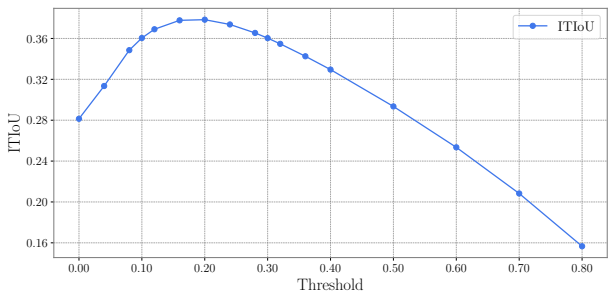


Figure 10: Impact of varying the threshold τ on $ITIoU$ performance for COCO-Stuff 27 dataset. The plot demonstrates the existence of an optimal τ, where segmentation performance is maximized.

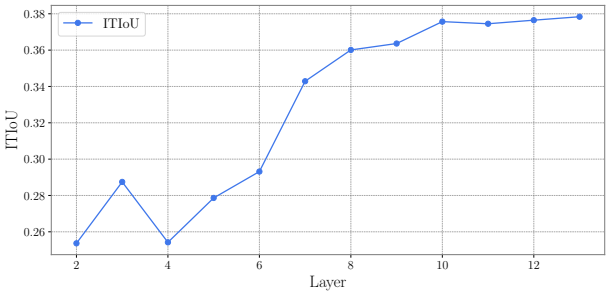


Figure 11: Layer-wise $ITIoU$ analysis for COCO-Stuff 27 dataset. Final layers perform significantly better than earlier ones due to their ability to capture high-level semantic features. This progression is evident in the increasing $ITIoU$ values.

E Threshold-Based Segmentation

Hierarchical clustering is used to segment the token explanation maps, where the threshold parameter ζ controls the level of granularity. Lower values of ζ produce fine-grained segmentations, while higher values merge similar clusters, leading to broader groupings.

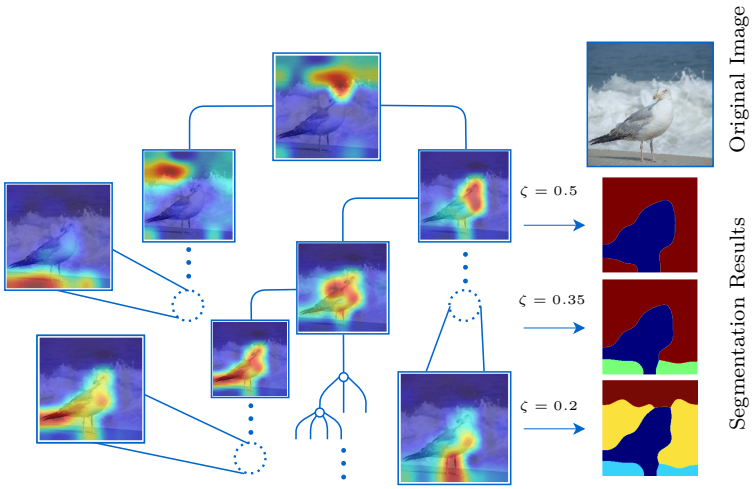


Figure 12: Hierarchical clustering tree showing the grouping of token explanation maps for all tokens in a latent layer of the Vision Transformer, not limited to the `CLS` token. Each leaf node represents a single token explanation map, while higher-level nodes show aggregated clusters based on a clustering threshold (ζ), which controls the level of detail. Lower ζ values reveal finer details, while higher values create broader, more general clusters.

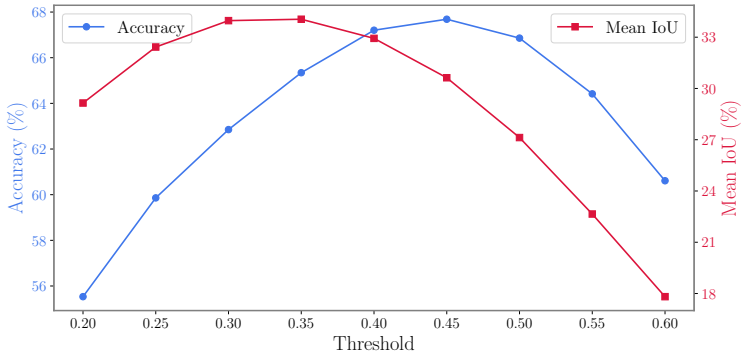


Figure 13: Impact of the clustering threshold ζ on segmentation performance for the COCO-Stuff 27 dataset. Accuracy and Mean IoU metrics show how different values of ζ affect segmentation quality.

Figure 12 illustrates the hierarchical clustering tree, where each leaf node represents a single token explanation map. As ζ increases, multiple explanation maps are grouped into larger clusters, reducing segmentation granularity. The rightmost part of the figure shows how different ζ values affect the final segmentation output.

Figure 13 further quantifies the impact of ζ on segmentation performance for the COCO-Stuff 27 dataset. Accuracy and Mean IoU metrics are plotted as a function of ζ, demonstrating that while different values of ζ yield varying levels of segmentation detail, our method remains robust across a range of threshold values.

The results confirm that selecting an appropriate ζ enables the model to segment objects at different levels of abstraction, demonstrating the adaptability of pre-trained Vision Transformers in unsupervised semantic segmentation. Table 6 summarizes the segmentation performance of our method across multiple datasets using a fixed threshold $\zeta = 0.4$. However, as discussed in Section B.1, not all threshold values may be equally suited for standard datasets.

Dataset	U. ACC	U. mIoU
COCO-Stuff 27	67.2	32.9
PASCAL VOC	-	51.9
Potsdam	74.6	-

Table 6: `ULTra` results across different datasets using the same threshold $\zeta = 0.4$. Accuracy (U. ACC) and Mean IoU (U. mIoU) are reported where available.

F Transformer Architecture

The architecture of a typical Transformer can be formulated as follows: the input X is split into n tokens $\{\mathbf{x}_i\}_{i=1}^n$. After tokenization, token embeddings $\{\mathbf{e}_i\}_{i=0}^n$ are computed, where $\mathbf{e}_0$ corresponds to the `CLS` token. Positional encodings PE_i are added to the i-th token embedding to incorporate spatial information, resulting in the latent token representation $\mathbf{z}_i^{(1)} = \mathbf{e}_i + \mathrm{PE}_i$. Here, $\mathbf{z}_i^{(l)}$ represents a latent token, where l denotes the layer index with $l \in \{1, \ldots, L\}$ and L is the total number of layers in the Transformer, and i represents the i-th token within the l-th layer.

For each head $h \in \{1, \ldots, H\}$ in the multi-head attention mechanism, the queries, keys, and values corresponding to the i-th token are obtained via linear transformations, projecting the latent token of dimension d into dimension k:

$$Q_h^{(l)}(\mathbf{z}_i^{(l-1)}) = (W_{h,q}^{(l)})^T \mathbf{z}_i^{(l-1)}, K_h^{(l)}(\mathbf{z}_i^{l-1}) = (W_{h,k}^{(l)})^T \mathbf{z}_i^{(l-1)},$$

$$V_h^{(l)}(\mathbf{z}_i^{(l-1)}) = (W_{h,v}^{(l)})^T \mathbf{z}_i^{(l-1)}, \quad \forall l \in \{2, \ldots, L\} \tag{16}$$

where $W_{h,q}^{(l)}, W_{h,k}^{(l)}, W_{h,v}^{(l)} \in \mathbb{R}^{d \times k}$. The attention weights for each token pair (i, j) at layer l and head h are computed as:

$$\alpha_{h,i,j}^{(l)} = \mathrm{softmax}_j \left(\frac{\langle Q_h^{(l)}(\mathbf{z}_i^{l-1}), K_h^{(l)}(\mathbf{z}_j^{(l-1)}) \rangle}{\sqrt{k}} \right). \tag{17}$$

Then, i-th token is updated by summing over the weighted values across all heads:

$$\bar{\mathbf{u}}_i^{(l)} = \sum_{h=1}^{H} (W_{c,h}^{(l)})^T \sum_{j=1}^{n} \alpha_{h,i,j}^{(l)} V_h^{(l)}(\mathbf{z}_j^{(l-1)}), \tag{18}$$

where $W_{c,h} \in \mathbb{R}^{k \times d}$. The updated token representation $\mathbf{u}_i$ after the attention layer is computed as:

$$\mathbf{u}_i^{(l)} = \mathrm{LayerNorm}(\mathbf{z}_i^{(l-1)} + \bar{\mathbf{u}}_i^{(l)}). \tag{19}$$

Each token then passes through a feed-forward network:

$$\bar{\mathbf{z}}_i^{(l)} = (W_2^{(l)})^T \mathrm{ReLU}((W_1^{(l)})^T \mathbf{u}_i), \tag{20}$$

$$\mathbf{z}_i^{(l)} = \mathrm{LayerNorm}(\mathbf{u}_i + \bar{\mathbf{z}}_i^{(l)}). \tag{21}$$

Here, $W_1^{(l)} \in \mathbb{R}^{d \times m}$, $W_2^{(l)} \in \mathbb{R}^{m \times d}$.