

# Beyond Labels: Self-Attention-Driven Semantic Separation Using Principal Component Clustering in Latent Diffusion Models

Felix Stillger<sup>1,2</sup><sup>a</sup>, Frederik Hasecke<sup>2</sup><sup>b</sup>, Lukas Hahn<sup>2</sup><sup>c</sup> and Tobias Meisen<sup>1</sup><sup>d</sup>

<sup>1</sup>University of Wuppertal, Gaußstraße 20, Wuppertal, Germany

<sup>2</sup>APTIV, Am Technologiepark 1, Wuppertal, Germany

{felix.stillger, meisen}@uni-wuppertal.de, {frederik.hasecke, lukas.hahn}@aptiv.com

Keywords: Diffusion Model, Self-Attention, Segmentation.

Abstract: High-quality annotated datasets are crucial for training semantic segmentation models, yet their manual creation and annotation are labor-intensive and costly. In this paper, we introduce a novel method for generating class-agnostic semantic segmentation masks by leveraging the self-attention maps of latent diffusion models, such as Stable Diffusion. Our approach is entirely learning-free and explores the potential of self-attention maps to produce semantically meaningful segmentation masks. Central to our method is the reduction of individual self-attention information to condense the essential features required for semantic distinction. We employ multiple instances of unsupervised k-means clustering to generate clusters, with increasing cluster counts leading to more specialized semantic abstraction. We evaluate our approach using state-of-the-art models such as Segment Anything (SAM) and Mask2Former, which are trained on extensive datasets of manually annotated masks. Our results, demonstrated on both synthetic and real-world images, show that our method generates high-resolution masks with adjustable granularity, relying solely on the intrinsic scene understanding of the latent diffusion model - without requiring any training or fine-tuning.


## 1 INTRODUCTION


Semantic segmentation is a fundamental task in computer vision, with applications ranging from autonomous driving to medical image analysis. However, the process of creating large, annotated datasets to train segmentation models is both time-consuming and costly. This has prompted increasing interest in methods that leverage existing data, models, or mechanisms to bypass the need for data creation and manual annotation. Generative models, particularly diffusion-based models like Stable Diffusion 2.1 (Rombach et al., 2022a), have shown remarkable capabilities in generating detailed and coherent images, yet their potential to assist in generating segmentation masks remains underexplored. In this work, we investigate the intrinsic ability of diffusion models to produce class-agnostic semantic segmentation masks without any modification to the models themselves or reliance on additional pre-trained networks (see Figure 1). Specifically, we exploit the self-attention





Figure 1: Our method generates class-agnostic yet semantically meaningful segmentation masks. The highlighted pixel (marked by a star in the upper left image) can be associated with various semantic categories, such as left eye, eyes, face, cat, and foreground. These segmentation masks are produced solely through the self-attention mechanism of Stable Diffusion, without relying on any external image features.

mechanisms embedded in latent diffusion models, which are designed to enhance image generation quality by capturing relationships between different parts of the image (Hong et al., 2023). While self-attention has been used in previous efforts to create segmenta-

<sup>a</sup> <https://orcid.org/0009-0006-9771-6233>

<sup>b</sup> <https://orcid.org/0000-0002-6724-5649>

<sup>c</sup> <https://orcid.org/0000-0003-0290-0371>

<sup>d</sup> <https://orcid.org/0000-0002-1969-559X>

tion masks, it has not been fully explored at the granularity of individual attention heads. We hypothesize that the self-attention heads within these models encode sufficient information about image structure and content, enabling the segmentation of distinct regions with semantically meaningful boundaries - without the need for external supervision.

Previous methods, such as (Nguyen et al., 2023) and (Tian et al., 2024), typically aggregate self-attention maps by averaging or summing over attention heads and/or features to manage the large tensor sizes involved. In contrast, our approach leverages the individual multi-head self-attention maps independently, preserving their distinct objectives and enabling the derivation of more fine-grained semantic masks.

Our main contributions are as follows:

- **Head-Wise Self-Attention Analysis.** We conduct a detailed analysis of the individual self-attention maps from each head in Stable Diffusion, demonstrating how they contribute to semantic separation within an image.
- **Class-Agnostic Mask Generation.** We propose a novel method for generating semantic segmentation masks across multiple levels of granularity—ranging from coarse to fine—directly from the self-attention features of the diffusion model.
- **Zero-Shot Segmentation.** We validate our approach in the context of zero-shot segmentation, showcasing the ability to interpret and semantically segment real-world images without any prior training or fine-tuning.

## 2 RELATED WORK

Numerous text-to-image diffusion models have been developed to generate images from textual prompts, with notable examples including DALL-E 3 (Betker et al., 2023), Imagen (Saharia et al., 2022), Muse (Chang et al., 2023), and Stable Diffusion (Rombach et al., 2022b). Among these, Stable Diffusion stands out as an open-source model capable of synthesizing high-resolution images containing multiple objects in one scene. This is achieved by encoding the input text into a latent space, where a diffusion process is applied using a denoising network. The final image is then reconstructed through a decoder.

Previous works have explored the role of self-attention in generative models, particularly diffusion-based models. For instance, (Vaswani et al., 2023) examined the self-attention mechanism in Stable Diffusion and concluded that it encapsulates valuable

layout and shape information. SegDiff (Amit et al., 2022) introduces a segmentation approach for diffusion models but relies on ground truth data for accurate segmentation. DiffuMask (Wu et al., 2023), building on AffinityNet (Ahn and Kwak, 2018), uses cross-attentions to generate foreground masks, yet, like SegDiff, it produces only one foreground mask per image. Dataset Diffusion (Nguyen et al., 2023) was the first to enable the generation of multiple object masks per image by combining both self- and cross-attentions. However, its reliance on cross-attentions results in coarse segmentation masks, as cross-attentions provide only broad region information and still require the finer details derived from self-attention maps.

SliME (Khani et al., 2024) refines self-attentions to improve cross-attention segmentation but still requires ground truth data to specify the segmentation style. Ref-Diff (Ni et al., 2023) demonstrates how generative models can leverage the connection between visual elements and text descriptions, and introduces a diffusional segmentor for zero-shot segmentation. DiffSeg (Tian et al., 2024) treats attention resolutions differently via an iterative merging approach but averages multi-head attentions, assuming similar objectives using Kullback-Leibler divergence. A network-free approach to obtain semantic segmentation masks is proposed in A network-free approach is proposed in (Feng et al., 2023), where semantic segmentation masks are generated by observing pixel connectivity through synthetic image variants, utilizing Generative Adversarial Networks (Goodfellow et al., 2014) rather than diffusion models. Meanwhile, iSeg (Sun et al., 2024) proposes an iterative refinement framework to reduce the entropy of self-attention maps, applying their module to unsupervised segmentation tasks and mask generation for synthetic datasets.

DAAM (Tang et al., 2022) introduces a method to visualize the cross-attention between words and pixels, producing pixel-level attribution maps using word-pixel scores from the denoising network. OVAM (Marcos-Manchón et al., 2024) extends this idea to generate cross-attention maps for open vocabulary tasks, enabling the segmentation of semantic meanings that may not be explicitly represented in the textual prompt without altering the image generation process.

To the best of our knowledge, no existing method leverages individual self-attention heads in a learning-free manner to generate multiple high-resolution masks for semantic separation within attention-based generative models.

### 3 INTERPRETING SELF-ATTENTIONS

The attention mechanism in diffusion models is used to focus on particular objectives and enhance reconstruction abilities, thereby improving the final sample quality (Hong et al., 2023). The basic scaled dot-product attention is computed as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

Where:  $Q \in \mathbb{R}^{n \times d_k}$  is the query matrix,  $K \in \mathbb{R}^{n \times d_k}$  is the key matrix,  $V \in \mathbb{R}^{n \times d_v}$  is the value matrix,  $d_k$  is the dimensionality of the key/query vectors,  $d_v$  the dimensionality of the value vector and  $n$  identical layers (Vaswani et al., 2023).

In multi-head attention, the final attention map is derived from the outputs of several individual attention heads. Each head processes the input by linearly projecting the query, key, and value matrices into distinct subspaces, allowing each head to focus on different parts of the input. These projections are then combined by concatenating the outputs from all heads, resulting in a richer representation that captures more nuanced relationships between the elements in the input.

$$\text{MHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (2)$$

with  $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$

Where:  $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$ ,  $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$ ,  $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ , and  $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$  (Vaswani et al., 2023).

This allows to jointly attend to information from different representations in different subspaces (Vaswani et al., 2023).

In the context of self-attention mechanisms, particularly within the context of multi-head attention, this means that each head in the transformer model is capable of focusing on distinct objectives and capturing different aspects of the features.

Stable Diffusion is comprised of three main components. To encode and decode images, a pre-trained VAE (Kingma and Welling, 2022) is employed. The second component is a text encoder, such as CLIP (Radford et al., 2021), which encodes a textual prompt into latent space. The core of the model is the denoising U-Net, which reconstructs images under conditioning from noise by learning a denoising strategy that incorporates self- and cross-attention mechanisms (Rombach et al., 2022b).

In Stable Diffusion, multi-head self-attention layers are positioned at various stages within the denoising

Table 1: Self-attention positions with layer name, count of individual heads, resolution and feature size of Stable Diffusion 2.1.

Name	Heads	Resolution	Feature Size
Down-1	5	4096 (64x64)	64
Down-2	10	1024 (32x32)	64
Down-3	20	256 (16x16)	64
Mid	20	64 (8x8)	64
Up-2	20	256 (16x16)	64
Up-3	10	1024 (32x32)	64
Up-4	5	4096 (64x64)	64

U-Net architecture. Our method utilizes Stable Diffusion 2.1, as this version features a larger number of attention heads compared to earlier versions, such as Stable Diffusion 1.5, which incorporates only eight parallel heads per layer (Rombach et al., 2022a). The specific positions and sizes of the self-attention layers in Stable Diffusion 2.1 are summarized in Table 1 and further detailed in Figure 12.

Previous methods for leveraging attention maps, such as the aforementioned Dataset Diffusion (Nguyen et al., 2023), aggregate information by summing and averaging the head-wise output over all time steps without using additional pre-trained models for practical purposes. The averaging and summing procedures lead to a loss of the individual separation capabilities inherent to each attention head, as they focus solely on aggregated areas of interest. To partially recover this lost precision, methods like Dataset Diffusion (Nguyen et al., 2023) average across multiple iteration steps. In contrast, our approach seeks to preserve and condense the rich information embedded within each self-attention map. We hypothesize that these maps inherently encode sufficient details to separate objects in the generated image based on their semantic meaning. Averaging across attention heads, however, can dilute the semantic information, limiting the ability to capture fine-grained distinctions.

Our methodology operates under the assumption that head-wise data can be effectively reduced to a more manageable form while preserving the unique subspace representation of each individual attention head. Furthermore, the specific objectives of each head can be visualized, allowing for direct inspection of their role in the segmentation process.

Each layer within the self-attention mechanism captures distinct features and serves different objectives. In Stable Diffusion 2.1, the number of attention heads varies based on their position within the architecture. The downstream flow contains two consecutive transformer layers for self-attention, while the upstream flow incorporates three consecutive transformer layers, and the middle block features a single transformer layer. Although the resolutions change in accordance

with the U-Net layers, the feature dimensions remain consistent across all layers.

Selecting the layers and features with the highest probabilities for object separation based on semantic meaning is a significant challenge due to the high-dimensional nature of the multi-head outputs. To facilitate visual interpretation of these objectives, we employ Principal Component Analysis (PCA) as a standard dimensionality reduction technique. A beneficial side effect of dimensionality reduction is its ability to highlight the most distinguishable elements of a feature map while simultaneously discarding noise. Visualizing the PCA of an individual attention head enables us to interpret and understand the semantics encoded within that head.

A straightforward method involves computing the top three principal components of a self-attention feature map for an individual head and mapping these components to RGB-values. In this mapping, similar colors indicate alignment across all three principal components, whereas visual differences in color highlight discrepancies in at least one component. This color-based separation allows for the easy identification of clusters and their corresponding objectives. Additionally, we interpret the resulting PCA values not only as a visual representation of the self-attention outputs but also as an encoding of objectives, with the distances between these representations reflecting similarities in image construction objectives. The resulting clusters/classes need not be human-understandable; they correspond to the intrinsic clusters that the model perceives as classes and their subclasses. Still the principal components are interpreted as important objectives that are able to represent a human-made class definition. This implies that a face contains subclasses like eyes, which are identified without prompting, but the model must understand that eyes must be in a particular area of the face. The PCA is conducted to understand the underlying objectives of each head and further to improve and accelerate downstream algorithms by extracting the most semantically significant data.

To illustrate our findings, we provide an example using the prompt: "a baby in a yellow toy car", along with the classes depicted in the image: ["person", "car"], follows the prompt style by (Nguyen et al., 2023), which appends the class names to the actual prompt separated by ";". This prompt is employed to generate an image using Stable Diffusion, and the resulting output is shown in Figure 2.

Figure 3 depicts the visualization of the first head of every multi-head attention (see Appendix Figure 12 for more information). The visualization starts with the downstream flow, transitioning from a 64x64



Figure 2: Output image by Stable Diffusion 2.1 based on the prompt: "a baby in a yellow toy car; person, car".

self-attention resolution to 16x16. As the resolution decreases, a greater number of distinct color clusters become apparent, while higher resolutions reveal finer clusters and more detailed features. The contours of objects are marked by abrupt color changes, effectively approximating the original image without relying on image space features. This demonstrates that the principal components successfully condense the objectives, visualize the objectives in a human-interpretable way, support our hypothesis of the intrinsic semantic information, and can be further utilized to segment an image into its semantic components.

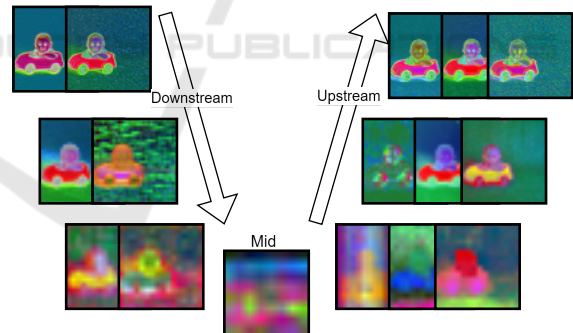


Figure 3: Principal component visualization of the first multi-head attention heads in U-Net order, progressing from left to right and top to bottom. The resolution begins at 64x64 and decreases to 16x16 in the down layers, with the mid block at 8x8 resolution. The upstream flow starts at 16x16 and increases back to 64x64 in the up layers. First three main components are mapped to RGB-values

The first self-attention in Figure 3, located at the upper left, demonstrates objective separation capability. The eyes, mouth, and hair are represented with distinct colors, indicating significant differences in at least one principal component, in comparison to the rest of the face and skin. It is noteworthy that the second (right) visualization of the first downstream layer

shows a distinct color cluster for the wheels (greenish) of the car, while the left head’s visualization has one color cluster for the black parts (including shadow of the car and wheels) of the car. Averaging over these head maps would result in the loss of important details, as some of these distinct clusters would be merged or diminished. This underscores the rationale for our approach of leveraging head-specific features, as individual heads may capture unique objectives that are otherwise lost through averaging.”

However, it is important to note that the car and the child’s face may appear closely related in this representation, leading to similar colors. This observation highlights the necessity of incorporating diverse heads and their respective features when applying clustering algorithms to ensure more effective separation of the image’s distinct semantic components.

The subsequent principal component visualization of the second down layer with a 64x64 resolution successfully separates the car from the child. However, as the resolution of the self-attention principal components decreases, the clusters lose finer details. Additionally, we observed subtle differences in semantic separation between the upstream and downstream self-attention layers, which we further analyze in Table 2.

Figure 3 already illustrates the advantage of visualizing individual heads. Notably, both the downstream and upstream layers reveal distinct color clusters that correspond well to the objects in the output image shown in Figure 2.

In addition to the upstream and downstream layers, Stable Diffusion 2.1 includes a single self-attention layer in the middle block. To explore its relevance, we conducted a PCA with three components and visualized the head outputs. The results revealed no noticeable color clusters corresponding to the semantic meaning of the output image, as shown in Figure 14, which includes all 20 heads of the middle block from the previous example.

Our experiments indicate that this layer does not contribute meaningful semantic information. We hypothesize that this lack of semantic separability is due to the block’s low resolution. This conclusion is further supported by the quantitative results presented in Table 8, where we repeated the analysis from Table 2 but included the middle self-attention features. The overall performance did not improve, and in some cases, it even worsened when the middle block features were incorporated. Consequently, we exclude the middle block from further investigation.

## 4 METHODOLOGY FOR MASK GENERATION

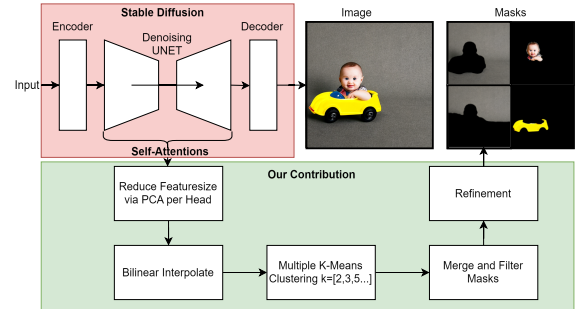


Figure 4: Main methodology of our proposed method.

Our method is built on top of the OVAM (Marcos-Manchón et al., 2024) source code to extract the raw self-attentions from the denoising U-Net. To demonstrate the effectiveness of our approach, we utilize publicly available text prompts from Dataset Diffusion (Nguyen et al., 2023), which are derived from images in the Pascal VOC training dataset (Everingham et al., 2012). The image captions for these prompts are generated using BLIP (Li et al., 2022).

### 4.1 Process Self-Attentions

Our main methodology, as illustrated in Figure 4, involves extracting self-attention maps from up to 16 different layers. The specific positions of these layers are detailed in Figure 12, with corresponding sizes presented in Table 1. To process the self-attention maps, we apply bilinear upsampling to the head-wise tensors, which are reduced via principal component analysis, ensuring a uniform shape across all layers. This approach yields PCA feature maps from up to 195 individual heads, all at the same resolution. Depending on their position within the network, these maps may capture different intrinsic semantic information.

To obtain masks from the reduced self-attentions, we apply k-means clustering to the stacked principal component features of the sub-selected heads. K-means is a simple clustering algorithm that does not need any additional parameters besides the cluster count  $k$ . This is an advantage if we want fast convergence for clusters without the need of fine-tuning parameters on our examples. We use a fixed seed for initialization to ensure that our method remains reproducible over all experiments. To address potential issues with fixed initialization, we experiment with various cluster counts and cluster the same features multiple times over an increasing cluster count. This

allows us to produce multiple masks with different semantic meaning per image. As a straightforward mask reduction step, we merge masks with a high intersection over union into a single unified mask. Examples of this approach are shown in Figure 5, where the generated images are displayed in the top row, and the corresponding clusters, with cluster counts  $\in [2, 5, 6, 10, 20]$ , are shown in the rows below. A low cluster count corresponds highly to a background-foreground segmentation [cluster count = 2] whereas the clusters with a cluster count of [5,6] correspond to high-level classes like child, car or cat. Clusters from a high cluster count [10, 20] correspond to low-level semantic segmentation masks like face, eyes and ears.

The resolution and the therefore abstraction level of the self-attentions is crucial: low-resolution self-attentions provide connectivity and distinguishable features for low-level objectives and therefore coarse segmentation but suffer from imprecise contours, whereas high-resolution self-attentions capture finer details, such as facial features, but may miss some obvious connections among masks (see Figure 11 for more details).

Depending on the target objective and computing time, parameters such as the *number of k-means clusters*, *upsampling resolution*, *number of principal components*, and position selection of the *self-attention layers* can be varied. We utilize scikit-learn (Pedregosa et al., 2011) to conduct our upsampling and clustering. The choice of the optimal setup depends on the desired outcome. For instance, if the goal is to segment only a person, a coarse segmentation, such as background versus foreground, might suffice. However, if the masks are intended for an image-to-image inpainting approach and a detailed definition of instance e.g. "eyes" are of interest, a much finer segmentation is required and therefore a higher cluster count has to be set.

As demonstrated qualitatively in Figure 5 and Figure 11, our approach is capable of segmenting fine-grained classes and multiple objects within a single scene. For instance, in the last column, a monitor displaying content is placed on a table in a room. Our method effectively separates the monitor, its content, and the table it is placed on. With an increasing cluster count, hierarchical masks can be defined, such as identifying a head as part of a person, which is part of the content on the monitor, which in turn belongs to the monitor. Additionally, in the Appendix Figure 11 provides a comparison of the generated clusters derived solely from the 16x16 self-attention maps.

This qualitative analysis shows that clusters derived from low-resolution self-attentions tend to capture more semantic meaning, closely aligning with

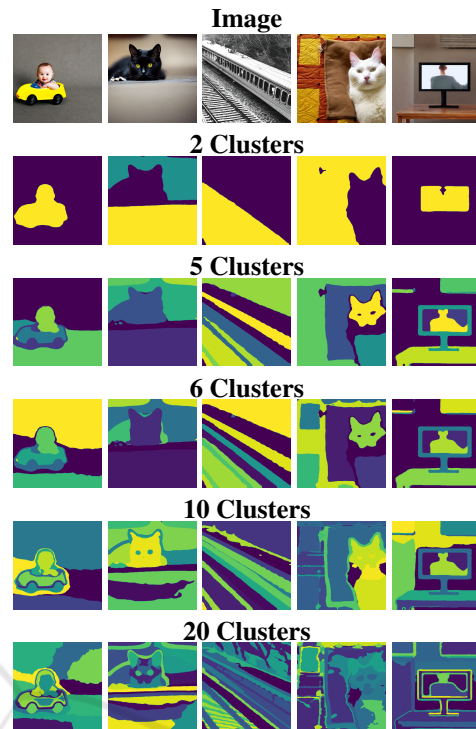


Figure 5: K-means clustering over multiple cluster counts and five examples. Every color represents a single cluster for attention-maps from sizes [16x16, 32x32, 64x64]. See more in Figure 11 (Appendix), where also clusters from only [16x16] are presented.

high-level classes such as those in Pascal VOC. In contrast, higher-resolution self-attentions are better suited for generating fine-grained, low-level segmentations. However, due to the significantly lower resolution of self-attentions - at minimum eight times lower than the output image — the principal components between clusters lack well-defined edges, especially in lower resolutions.

As the number of clusters increases, some pseudo-clusters may form due to transitions between truly meaningful clusters, which is an inherent limitation of the bilinear interpolation process. Additionally, object contours remain uncertain because no final output image features are incorporated into the clustering process. Introducing extraneous information, such as pixel positions or image features, could result in clusters being overly influenced by these features, thereby reducing the interpretability and effectiveness of the clustering method.

To improve accuracy, a refinement method is needed. We propose leveraging the upsampling error as a potential means of refinement and incorporating image features in a post-processing step. A brief analysis of the upsampling error is provided in the Appendix 6.

## 4.2 Hyperparameter-Studies and Evaluation

In order to also quantitatively evaluate our method and determine suitable parameters, we use two external task-specific state-of-the-art models to generate pseudo-labels on images generated by the diffusion model. First, we employ the Segment Anything model (Kirillov et al., 2023) with a ViT-L backbone, which is expected to generate instance masks without domain knowledge. Second, we use Mask2Former (Cheng et al., 2022) with a Swin-L backbone, which is pre-trained on the ADE20K dataset (Zhou et al., 2017) and is available as a pre-trained implementation via the mmseg toolbox (Contributors, 2020). Unlike instance segmentation, this model provides class-specific semantic segmentations, and its training classes closely align with the Pascal VOC classes, which are the basis for the prompts used in our model.

Both of these methods are trained on labeled segmentation ground truth, whereas our method has never seen or trained on a segmentation mask. Instead, our method relies solely on the intrinsic semantic knowledge derived from the self-attentions of Stable Diffusion and its derived features.

To assess the performance of our method, we compare the semantic masks generated by our approach with those produced by the two evaluation models. This comparison aims to highlight the effectiveness of our method relative to an instance-focused model and a class-based segmentation model. We omit the classification component, as it would only be feasible with cross-attention. For evaluation, we use Intersection over Union (IoU) as our metric, defined as follows:

$$\text{IoU}(A,B) = \frac{\text{Area of Intersection}}{\text{Area of Union}} = \frac{|A \cap B|}{|A \cup B|} \quad (3)$$

To provide a comprehensive evaluation of the most significant masks, we define the *Top-n* metric. This metric selects the  $n$ -largest masks from the external models and computes the average IoU with the corresponding classes from our method:

$$\text{Top-n} = \frac{1}{N} \sum_{n=1}^N \max_{B \in \text{predicted masks}} \left( \frac{|A_n \cap B|}{|A_n \cup B|} \right) \quad (4)$$

where  $A_n$  represents the  $n$ -th largest mask from the external models, and  $B$  denotes the masks from our method.

For the final evaluation, we average this score over all samples. Figure 6 presents our evaluation for the first sample using the Segment Anything Model (SAM), and Figure 7 shows the performance of our method compared to Mask2Former.

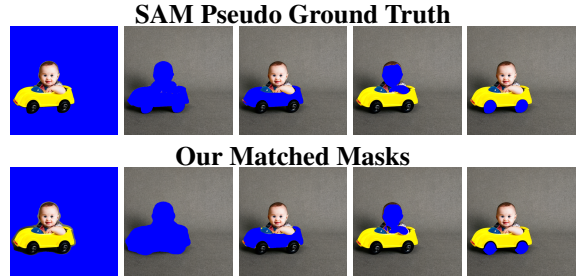
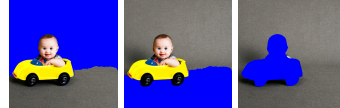


Figure 6: Top: SAM Top-5 masks colored in blue. Bottom: Matched masks from our method. The IoUs between the pseudo ground truth and our method’s masks are 0.97, 0.82, 0.94, 0.87, and 0.77 (from left to right).

The SAM evaluation provides an example of fine, pixel-accurate masks. While it segments parts of the image very effectively, our method identifies not only the corresponding masks but also additional semantically consistent variations that are not identified by SAM. We do not provide additional domain knowledge to SAM, such as masks, points or boxes, to obtain the raw scene-interpreted masks from SAM. On the other hand, the Mask2Former evaluation serves as a sanity check, as it was trained on the ADE20K dataset, which limits its ability to generalize beyond its training domain. In the example, Mask2Former splits the background into upper and one lower parts, with only a single mask for the foreground. Our method closely matches these masks of the Mask2Former network as shown in Figure 7.

It is important to note that SAM contributes to the Top-1 to Top-5 average score due to its ability to generate a larger number of masks. In contrast, Mask2Former only contributes to the Top-1 to Top-3 average score, as it generates fewer class-specific masks. We observed that SAM typically produces more masks than Mask2Former, making it a more reliable source for the Top-n metric.

Mask2Former Pseudo Ground Truth



Our Matched Masks

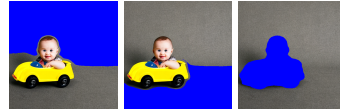


Figure 7: Top: Mask2Former Top-3 masks colored in blue. Bottom: Matched masks from our method. The IoUs between the pseudo ground truth and our method’s masks are 0.90, 0.87, and 0.82 (from left to right).

Next, we present a study on layer positions and reweighting of the features in Table 2. We hypothesize that the position of the layer affects the semantic

Table 2: Study on reweighting and layer positions, evaluated by the average top scores across multiple samples (measured in average IoU). Self-attentions at resolutions of [16, 32, 64] are used with three principal components, bilinearly upsampled to a resolution of 64. No middle block is incorporated.

ID	Reweight	Layer	SAM Top-1	SAM Top-3	SAM Top-5	M2F Top-1	M2F Top-3	M2F Top-5
0	True	down	0.80	0.79	0.76	0.76	0.73	0.64
1	True	up	0.84	0.82	0.79	0.79	0.76	0.67
2	True	both	0.84	0.81	0.78	0.79	0.76	0.66
6	False	down	0.84	0.82	0.79	0.80	<b>0.78</b>	0.67
7	False	up	<b>0.85</b>	<b>0.83</b>	<b>0.80</b>	<b>0.81</b>	<b>0.78</b>	<b>0.68</b>
8	False	both	0.84	0.83	<b>0.80</b>	0.80	<b>0.78</b>	<b>0.68</b>

meaning of the generated masks. While the down-flow is expected to focus on condensing information, the up-flow is hypothesized to support image reconstruction. The reweighting process addresses the imbalance in the feature map distribution across different layer positions, particularly due to the higher number of 16x16 attention layers compared to 64x64 layers. The reweighting is defined as follows:

$$x_{reweighted} = \left[ \frac{x_{original}(r_i) \cdot r_i}{\max(r_1, r_2, r_3)} \right]_{r_i \in \{16, 32, 64\}} \quad (5)$$

where  $x$  are the features which get reweighted by their resolution.

The study presented in Table 2 demonstrates that self-attentions from the upstream layers are more valuable for reconstructing accurate masks compared to the downstream layers. All rows with only "down"-layers are inferior to gather only "up"-layers for self-attention. However, including features from the downstream layers does not disadvantage the performance, but using "both"-self-attentions has no effect on performance. Also, reweighting has no positive impact in this scenario, and the non-reweighted clusters are slightly ahead, in conclusion supporting the importance of the low-resolution self-attention maps.

We also conducted a study on the impact of the number of principal components, as shown in Figures 8 and Table 7. These figures demonstrate that applying PCA to the self-attention features enhances clustering compared to using the raw features. While there is no significant change in mask accuracy, the speedup achieved is defined by  $speedup = \frac{64}{\#principal\ components}$ . The study shows an improvement in the SAM Top-n metric when using approximately eight principal components, with the optimal number being around 32. This analysis was conducted with bilinear feature upsampling to a resolution of 256. The study also reveals that we can reduce the original feature parameters by more than a factor of 20 (by applying PCA to eight components and using only the "up" attention layers) without sacrificing the final average performance.

Finally, we examined how the position of feature layers affects which resolution of self-attention pro-

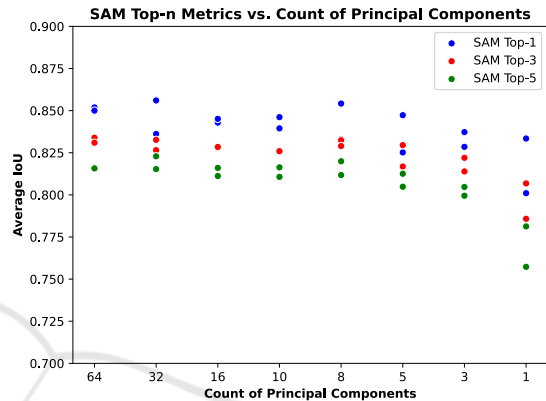


Figure 8: Impact of varying the number of principal component on multiple SAM-scores. Higher dimensionality does not necessarily lead to a greater performance. Detailed metrics in Table 7.

vides the most accurate semantic information. This is motivated by the fact that we have qualitatively noticed that the low-resolution self-attention features represent low-level semantics, while the higher-resolution self-attention carries finer details (see Figure 11). Conversely, the higher resolution self-attentions should tend to better represent the contour of an object because of the higher feature resolution. Based on this study, we concluded that the PCA components should be taken from all reasonable attention layers across all resolutions, specifically the [16x16, 32x32, 64x64] self-attention layers for best performance, as shown in Table 6. The mix of coarse and fine features provides the best performance. For efficiency, we bilinearly upsampled to 64x64 to perform this study in a reasonable time frame.

## 5 ZERO-SHOT SEGMENTATION

To demonstrate the capabilities of our findings, we conduct our method with an image-to-image approach to enable zero-shot segmentation on all types of images. To validate, we apply our method on a real-world image, which we try to denoise with Sta-



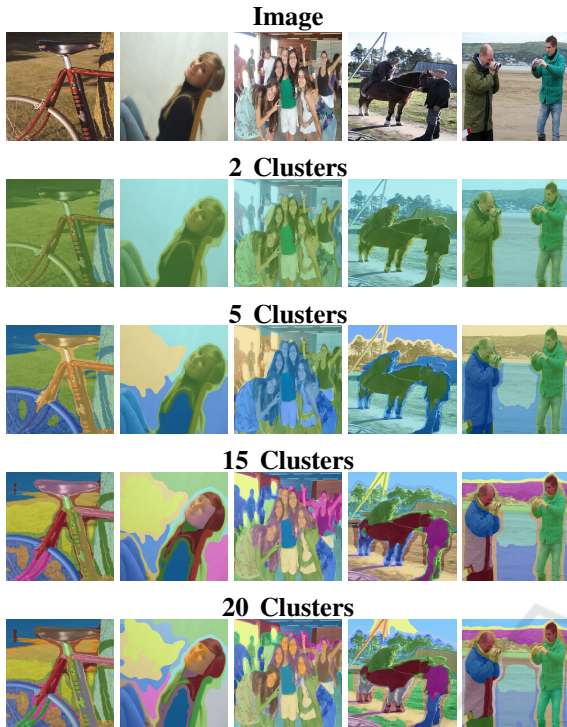


Figure 9: Zero-Shot segmentation of our method with ascending cluster counts from top to bottom, going from high-level segmentation to low-level segmentation.

ble Diffusion to obtain the self-attention features. In order not to change the original content of the image too much and still obtain segmentation masks we apply only a small denoising strength ( $strength = 0.05$ ) and only perform 2 iteration steps. Furthermore, we do not use an input prompt (we input an empty string), but one can easily be generated with any image descriptor, such as BLIP (Li et al., 2022). We validate our method using the public datasets Pascal VOC and Cityscapes, where we are able to compare against a public ground truth and the performance of the SOTA-segmentation algorithm SAM. Our workflow is the same as presented previously in section 4. Figure 9 illustrates the general capabilities of our method on real-world images. Our method demonstrates that self-attention features convey semantic information for classes and even subclasses, and that the internal model’s interpretation of them can be visualized. The zero-shot application proves that the self-attentions can be employed to segment the content of an image not only working for image generation by textual prompt. This method can also be utilized to obtain segmentation masks in the wild. Remarkable in our method is that it connects same classes across pixel gaps and group classes together, visualized in the third image of Figure 9 from the left. People’s faces or clothes are assigned to the same cluster.

Table 3: Performance comparison of our method and SAM variants on Cityscapes and Pascal VOC validation datasets in average class IoU.

Validation Set	Our Method	SAM ViT-H	SAM ViT-B	SAM ViT-L
Pascal VOC Semantic	0.66	<b>0.69</b>	0.39	0.65
Pascal VOC Instance	0.62	<b>0.74</b>	0.39	0.65
Cityscapes Semantic	0.27	<b>0.44</b>	0.26	0.43
Cityscapes Instance	0.21	<b>0.26</b>	0.15	0.25

As the number of clusters increases, these high-level clusters split, as discussed, into lower-level clusters. As the number of clusters increases, the high-level clusters are divided into lower-level clusters. This is possible due to the lack of additional pixel position data or additional information.

In order to evaluate and compare our method with SAM, we conduct an analysis using the Pascal VOC and Cityscapes validation sets. We bypass the classification task and match the generated mask with the highest IoU to the ground truth mask for both SAM and our method. Subsequently, we determined the class-wise IoU per image and averaged the results over all samples. We chose an average class IoU per image instead of an mIoU to not consider complex scenes with finer and smaller masks less than simple scenes. We use both the semantic and instance labels for comparison. The results are presented in Table 3, which shows that the best performance was achieved on the semantic segmentation validation set of the Pascal VOC dataset. Our method outperforms the strongest SAM with ViT-H backbone in some classes (see Appendix Table 4 for more details).

However, the overall average class performance is slightly below that of the best performing SAM model, but above the rest. The discrepancy widens in the case of instance segmentation, where the metric of the best performing SAM model is noticeably higher, but our method is close to the SAM model with ViT-L backbone. In the Cityscapes datasets, there are a lot more finer and complex segmentations, and our method falls slightly more behind compared to best performance of the best SAM and can now only perform better than the weakest SAM model with ViT-B backbone.

Nevertheless, our performance is remarkable considering that we use an untrained method for semantic segmentation, while SAM is self-supervised and trained on billions of images and millions of annotated masks. We have added some examples in the Appendix in Figure 13 where one will notice that our method still cannot match the pixel accuracy of SAM, but is able to obtain masks where a class is well represented. Furthermore, our method is superior in some edge cases where a class is occluded by other things and our method is able to associate a class with a mask where all SAM model fail to segment (see Figure 13 example horse). This is due to the underlying

scene understanding of the Stable Diffusion model compared to a mask-generating optimized model such as SAM.

## 6 CONCLUSION

We have presented a novel approach to generate high-resolution semantic masks using only the self-attention maps from diffusion models. We show that our method extracts semantically meaningful masks, without requiring additional learning or pre-trained models. This approach can be employed to directly obtain semantic masks for self-generated images using textual prompts as input only or for zero-shot segmentation, where an input image is given. Our method enables the utilization of Stable Diffusion’s inherent scene understanding for semantic separation, a task it has not been explicitly trained on. Validation with SAM (Kirillov et al., 2023) shows that our approach produces high-quality semantic segmentation on par with state-of-the-art methods, further allowing flexibility to adjust segmentation granularity. We further show that the generated masks benefit from Stable Diffusion’s scene understanding, to provide clusters of consistent semantic meaning beyond occlusion and pixel gaps.

As a direction for future work, we suggest using cross-attention maps to further obtain class labels for the generated masks. Additionally, a post-processing step using the upsampling error and image features to refine the masks will increase the accuracy of the masks.

## REFERENCES

- Ahn, J. and Kwak, S. (2018). Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation.
- Amit, T., Shaharbany, T., Nachmani, E., and Wolf, L. (2022). Segdiff: Image segmentation with diffusion probabilistic models.
- Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., et al. (2023). Improving image generation with better captions. *Computer Science*. <https://cdn.openai.com/papers/dall-e-3.pdf>, 2(3):8.
- Chang, H., Zhang, H., Barber, J., Maschinot, A., Lezama, J., Jiang, L., Yang, M.-H., Murphy, K., Freeman, W. T., Rubinstein, M., Li, Y., and Krishnan, D. (2023). Muse: Text-to-image generation via masked generative transformers.
- Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., and Girdhar, R. (2022). Masked-attention mask transformer for universal image segmentation.
- Contributors, M. (2020). MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. <https://github.com/open-mmlab/mms Segmentation>.
- Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2012). The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>.
- Feng, Q., Gadde, R., Liao, W., Ramon, E., and Martinez, A. (2023). Network-free, unsupervised semantic segmentation with synthetic images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23602–23610.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks.
- Hong, S., Lee, G., Jang, W., and Kim, S. (2023). Improving sample quality of diffusion models using self-attention guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7462–7471.
- Khani, A., Taghanaki, S. A., Sanghi, A., Amiri, A. M., and Hamarneh, G. (2024). Slime: Segment like me.
- Kingma, D. P. and Welling, M. (2022). Auto-encoding variational bayes.
- Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., and Girshick, R. (2023). Segment anything.
- Li, J., Li, D., Xiong, C., and Hoi, S. (2022). Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation.
- Marcos-Manchón, P., Alcover-Couso, R., SanMiguel, J. C., and Martínez, J. M. (2024). Open-vocabulary attention maps with token optimization for semantic segmentation in diffusion models.
- Nguyen, Q., Vu, T., Tran, A., and Nguyen, K. (2023). Dataset diffusion: Diffusion-based synthetic dataset generation for pixel-level semantic segmentation.
- Ni, M., Zhang, Y., Feng, K., Li, X., Guo, Y., and Zuo, W. (2023). Ref-diff: Zero-shot referring image segmentation with generative models. *arXiv preprint arXiv:2308.16777*.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision.
- Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022a). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695.

- Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022b). High-resolution image synthesis with latent diffusion models.
- Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., Salimans, T., Ho, J., Fleet, D. J., and Norouzi, M. (2022). Photorealistic text-to-image diffusion models with deep language understanding.
- Sun, L., Cao, J., Xie, J., Khan, F. S., and Pang, Y. (2024). iseg: An iterative refinement-based framework for training-free segmentation.
- Tang, R., Liu, L., Pandey, A., Jiang, Z., Yang, G., Kumar, K., Stenetorp, P., Lin, J., and Ture, F. (2022). What the daam: Interpreting stable diffusion using cross attention.
- Tian, J., Aggarwal, L., Colaco, A., Kira, Z., and Gonzalez-Franco, M. (2024). Diffuse attend and segment: Unsupervised zero-shot segmentation using stable diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3554–3563.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2023). Attention is all you need.
- Wu, W., Zhao, Y., Shou, M. Z., Zhou, H., and Shen, C. (2023). Diffumask: Synthesizing images with pixel-level annotations for semantic segmentation using diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1206–1217.
- Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. (2017). Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641.

## APPENDIX

**Upsampling Error.** The upsampling effect/error can be computed as follows:

$$E_{Upsample}(i, j) = \frac{1}{n} \sum_{k=1}^n \sqrt{(x_{bilinear}(i, j, k) - x_{nearest}(i, j, k))^2} \quad (6)$$

$\forall i \in \{1, 2, \dots, \text{width}\}, \forall j \in \{1, 2, \dots, \text{height}\}$   
 $n = \text{count of principal components}$

This calculation computes a pixel-wise mean of the root squared differences of the nearest neighbor ( $x_{nearest}$ ) and bilinear upsampled ( $x_{bilinear}$ ) principal components. This approach facilitates a visual estimation of clusters, as illustrated in Figure 10. The upsampling effect can be leveraged for contour refinement of the final masks, filtering out potential pseudo-clusters, or identifying objects in a generated image.

Table 4: Our Method vs. SAM on the Pascal VOC validation set for **semantic** segmentation. The metric is the per image averaged class-wise Iou.

Class	Our Method	SAM ViT-H	SAM ViT-L	SAM ViT-B
airplane	0.65	<b>0.73</b>	0.70	0.34
bicycle	<b>0.30</b>	0.26	0.23	0.14
bird	0.70	<b>0.79</b>	0.74	0.47
bottle	0.52	<b>0.74</b>	0.72	0.52
bus	0.79	<b>0.85</b>	0.85	0.49
car	0.61	<b>0.77</b>	0.74	0.54
cat	0.83	<b>0.86</b>	0.84	0.46
chair	0.52	<b>0.56</b>	0.53	0.40
table	<b>0.71</b>	0.55	0.49	0.36
dog	0.79	<b>0.84</b>	0.83	0.39
horse	<b>0.70</b>	0.65	0.60	0.20
motorbike	<b>0.67</b>	0.66	0.50	0.18
person	0.61	<b>0.64</b>	0.63	0.39
potted plant	0.42	<b>0.44</b>	0.42	0.30
sheep	<b>0.73</b>	0.64	0.63	0.50
sofa	<b>0.80</b>	0.67	0.62	0.43
train	0.79	<b>0.82</b>	0.78	0.16
monitor	0.60	<b>0.83</b>	0.82	0.68
all classes	0.66	<b>0.69</b>	0.65	0.39

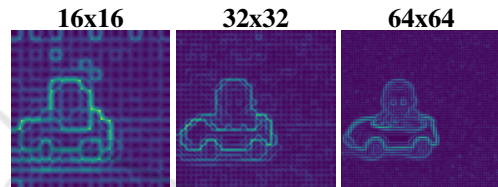


Figure 10: Pixel-mean upsampling effect of individual resolutions of the self-attentions for the "baby"-example.

Table 5: Our Method versus SAM on the Pascal VOC validation set for **instance** segmentation. The metric is the per mask averaged class-wise IoU. The differences compared to semantic segmentation for Pascal VOC are minimal because multiple instances of a single object are rare in this dataset.

Class	Our Method	SAM ViT-H	SAM ViT-L	SAM ViT-B
airplane	0.62	<b>0.73</b>	0.70	0.34
bicycle	<b>0.29</b>	0.28	0.24	0.14
bird	0.69	<b>0.84</b>	0.74	0.47
bottle	0.48	<b>0.81</b>	0.72	0.52
bus	0.70	<b>0.92</b>	0.85	0.49
car	0.54	<b>0.81</b>	0.74	0.54
cat	0.81	<b>0.89</b>	0.84	0.46
chair	0.48	<b>0.65</b>	0.53	0.40
table	<b>0.69</b>	0.56	0.49	0.36
dog	0.77	<b>0.88</b>	0.83	0.39
horse	0.67	<b>0.70</b>	0.60	0.20
motorbike	0.63	<b>0.68</b>	0.50	0.18
person	0.54	<b>0.75</b>	0.63	0.39
potted plant	0.40	<b>0.54</b>	0.42	0.30
sheep	0.60	<b>0.79</b>	0.63	0.50
sofa	<b>0.79</b>	0.71	0.62	0.43
train	0.78	<b>0.83</b>	0.78	0.16
monitor	0.59	<b>0.91</b>	0.82	0.68
all classes	0.62	<b>0.74</b>	0.65	0.39

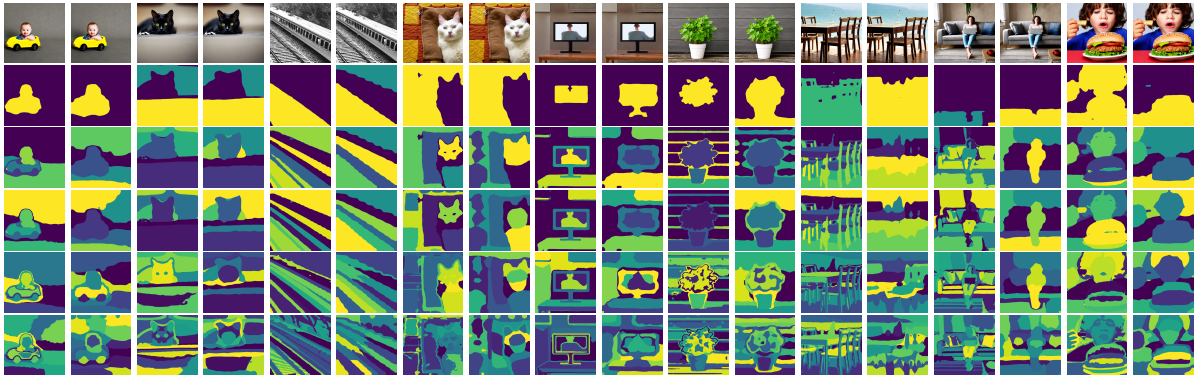


Figure 11: Examples comparing high-resolution masks from [16x16, 32x32, 64x64] attentions (left) with coarse masks from [16x16] attentions (right). High-resolution masks capture finer details and sharper contours, while the [16x16] masks lack precision in object boundaries, but focus more on high-level semantic meanings.

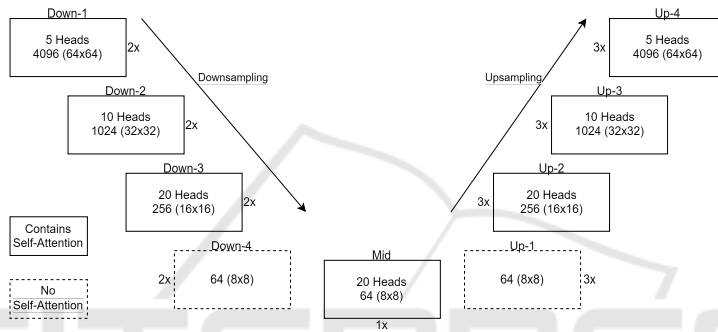


Figure 12: Positions of self-attentions in denoising U-Net in Stable Diffusion 2.1. There are six self-attention in the down layers, one in the middle layer and 9 in the up layers, down-4 and up-1 does not contain a transformer layer.

Table 6: Parameter study on attention layers position and principal component count, only with 64x64 bilinear feature upsampling.

ID	SAM Top-1	SAM Top-3	SAM Top-5	SAM Top-10	SAM Top-15	M2F Top-1	M2F Top-2	M2F Top-3	M2F Top-5	M2F Top-10	Reweight	Attention Resolution	#Principal Components	Upsample Resolution	Sample Count
0	0.84	0.83	0.80	0.73	0.67	0.79	0.79	0.77	0.68	0.59	True	[16, 32, 64]	64	64	50
1	0.84	0.82	0.80	0.72	0.67	0.79	0.78	0.76	0.67	0.59	True	[16, 32, 64]	32	64	50
2	0.84	0.82	0.79	0.72	0.66	0.80	0.79	0.77	0.68	0.60	True	[16, 32, 64]	16	64	50
3	0.84	0.82	0.80	0.72	0.67	0.79	0.78	0.76	0.68	0.60	True	[16, 32, 64]	10	64	50
4	0.83	0.82	0.79	0.71	0.66	0.78	0.78	0.76	0.67	0.59	True	[16, 32, 64]	8	64	50
5	0.84	0.82	0.79	0.70	0.65	0.79	0.78	0.76	0.67	0.59	True	[16, 32, 64]	5	64	50
6	0.84	0.81	0.78	0.70	0.65	0.77	0.76	0.70	0.66	0.56	True	[16, 32, 64]	3	64	50
7	0.82	0.80	0.76	0.67	0.62	0.77	0.76	0.74	0.65	0.54	True	[16, 32, 64]	1	64	50
8	0.83	0.82	0.80	0.72	0.67	0.78	0.77	0.76	0.67	0.59	True	[32, 64]	64	64	50
9	0.82	0.81	0.79	0.72	0.67	0.78	0.76	0.75	0.66	0.58	True	[32, 64]	32	64	50
10	0.82	0.81	0.79	0.71	0.66	0.78	0.76	0.75	0.66	0.58	True	[32, 64]	16	64	50
11	0.83	0.81	0.79	0.71	0.67	0.78	0.76	0.75	0.67	0.59	True	[32, 64]	10	64	50
12	0.82	0.81	0.78	0.71	0.66	0.77	0.76	0.75	0.66	0.59	True	[32, 64]	8	64	50
13	0.83	0.81	0.78	0.70	0.66	0.76	0.75	0.74	0.66	0.58	True	[32, 64]	5	64	50
14	0.82	0.81	0.78	0.69	0.65	0.77	0.75	0.74	0.65	0.57	True	[32, 64]	3	64	50
15	0.80	0.78	0.73	0.66	0.62	0.75	0.73	0.71	0.62	0.53	True	[32, 64]	1	64	50
16	0.84	0.81	0.79	0.70	0.62	0.80	0.80	0.78	0.68	0.60	True	[16, 32]	64	64	50
17	0.84	0.82	0.79	0.69	0.62	0.80	0.80	0.78	0.68	0.60	True	[16, 32]	32	64	50
18	0.84	0.82	0.79	0.70	0.62	0.79	0.80	0.78	0.69	0.59	True	[16, 32]	16	64	50
19	0.84	0.82	0.79	0.70	0.62	0.80	0.80	0.78	0.68	0.59	True	[16, 32]	10	64	50
20	0.84	0.81	0.78	0.70	0.62	0.79	0.80	0.78	0.68	0.59	True	[16, 32]	8	64	50
21	0.84	0.81	0.78	0.69	0.62	0.79	0.80	0.78	0.68	0.58	True	[16, 32]	5	64	50
22	0.83	0.81	0.78	0.69	0.61	0.79	0.79	0.77	0.68	0.58	True	[16, 32]	3	64	50
23	0.82	0.80	0.76	0.67	0.60	0.77	0.78	0.76	0.66	0.57	True	[16, 32]	1	64	50
24	0.82	0.81	0.79	0.71	0.66	0.78	0.76	0.75	0.66	0.58	True	[64]	64	64	50
25	0.82	0.80	0.79	0.71	0.66	0.77	0.76	0.74	0.65	0.58	True	[64]	32	64	50
26	0.82	0.80	0.78	0.70	0.66	0.77	0.76	0.75	0.66	0.58	True	[64]	16	64	50
27	0.82	0.80	0.78	0.70	0.66	0.77	0.76	0.74	0.66	0.58	True	[64]	10	64	50
28	0.81	0.80	0.78	0.70	0.66	0.77	0.76	0.74	0.66	0.57	True	[64]	8	64	50
29	0.81	0.79	0.77	0.69	0.65	0.76	0.75	0.74	0.65	0.56	True	[64]	5	64	50
30	0.81	0.79	0.76	0.69	0.64	0.76	0.75	0.73	0.65	0.56	True	[64]	3	64	50
31	0.79	0.78	0.74	0.66	0.61	0.74	0.73	0.71	0.63	0.54	True	[64]	1	64	50

Table 7: Study on principal component count with 256x256 bilinear feature upsampling.

ID	SAM Top-1	SAM Top-3	SAM Top-5	SAM Top-10	SAM Top-15	M2F Top-1	M2F Top-2	M2F Top-3	M2F Top-5	M2F Top-10	Reweight	Attention Resolution	#Principal Components	Upsample Resolution	Sample Count
0	0.85	0.83	0.82	0.74	0.69	0.80	0.79	0.78	0.69	0.61	True	[16, 32, 64]	64	256	41
1	0.86	0.83	0.82	0.75	0.70	0.79	0.79	0.78	0.68	0.62	True	[16, 32, 64]	32	256	41
2	0.84	0.83	0.82	0.74	0.69	0.79	0.78	0.77	0.69	0.63	True	[16, 32, 64]	16	256	41
3	0.85	0.83	0.82	0.74	0.69	0.80	0.79	0.77	0.68	0.61	True	[16, 32, 64]	10	256	41
4	0.85	0.83	0.82	0.74	0.68	0.79	0.79	0.78	0.68	0.61	True	[16, 32, 64]	8	256	41
5	0.85	0.83	0.81	0.74	0.68	0.79	0.78	0.77	0.68	0.59	True	[16, 32, 64]	5	256	41
6	0.84	0.82	0.81	0.73	0.68	0.78	0.77	0.76	0.67	0.60	True	[16, 32, 64]	3	256	41
7	0.83	0.81	0.78	0.69	0.64	0.77	0.76	0.74	0.65	0.57	True	[16, 32, 64]	1	256	41
8	0.85	0.83	0.82	0.75	0.69	0.78	0.78	0.76	0.68	0.60	True	[32, 64]	64	256	41
9	0.84	0.83	0.82	0.75	0.70	0.78	0.78	0.77	0.68	0.62	True	[32, 64]	32	256	41
10	0.85	0.83	0.81	0.75	0.70	0.79	0.78	0.77	0.68	0.61	True	[32, 64]	16	256	41
11	0.84	0.83	0.81	0.74	0.70	0.79	0.78	0.76	0.68	0.61	True	[32, 64]	10	256	41
12	0.83	0.83	0.81	0.74	0.70	0.77	0.77	0.76	0.67	0.60	True	[32, 64]	8	256	41
13	0.83	0.82	0.81	0.73	0.69	0.77	0.76	0.76	0.67	0.61	True	[32, 64]	5	256	41
14	0.83	0.81	0.80	0.73	0.68	0.76	0.76	0.75	0.66	0.60	True	[32, 64]	3	256	41
15	0.80	0.79	0.76	0.68	0.64	0.74	0.73	0.72	0.63	0.56	True	[32, 64]	1	256	41

Table 8: Incorporating middle self-attention and study on reweighting and layer positions (like Table 2), evaluated by the average of the top scores over multiple samples (in average IoU). [8,16,32,64] self-attention layers were used with 3 principal components, feature bilinear upsampling resolution to 64 and additionally middle block is incorporated.

ID	Reweight	Layer	SAM Top 1	SAM Top 3	SAM Top 5	M2F Top-1	M2F Top-3	M2F Top-5
0	True	down+mid	0.80	0.78	0.75	0.75	0.73	0.64
1	True	up+mid	0.84	0.82	0.79	0.79	0.77	0.67
2	True	all+mid	0.84	0.82	0.79	0.79	0.76	0.67
3	False	down+mid	0.83	0.81	0.78	0.79	0.77	0.67
4	False	up+mid	<b>0.85</b>	0.82	0.79	<b>0.80</b>	<b>0.78</b>	<b>0.68</b>
5	False	all+mid	0.84	<b>0.83</b>	<b>0.80</b>	<b>0.80</b>	<b>0.78</b>	<b>0.68</b>



Figure 13: Comparison of segmentation performance between SAM models and our method on the Pascal VOC validation set.

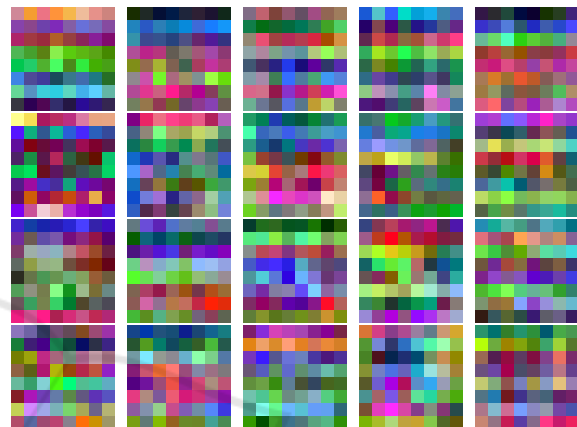


Figure 14: Visualization of the self-attentions from the mid-block for the "baby"-example presented in Section 3. No distinct clusters are noticeable when mapping the three main components to RGB-values.