Peekaboo: Text to Image Diffusion Models are Zero-Shot Segmentors

Ryan Burgert  Kanchana Ranasinghe  Xiang Li  Michael S. Ryoo
Stony Brook University
rburgert@cs.stonybrook.edu

Abstract

Recent diffusion-based generative models combined with vision-language models are capable of creating realistic images from natural language prompts. While these models are trained on large internet-scale datasets, such pre-trained models are not directly introduced to any semantic localization or grounding. Most current approaches for localization or grounding rely on human-annotated localization information in the form of bounding boxes or segmentation masks. The exceptions are a few unsupervised methods that utilize architectures or loss functions geared towards localization, but they need to be trained separately.

In this work, we explore how off-the-shelf diffusion models, trained with no exposure to such localization information, are capable of grounding various semantic phrases with no segmentation-specific re-training. An inference time optimization process is introduced, that is capable of generating segmentation masks conditioned on natural language. We evaluate our proposal Peekaboo for unsupervised semantic segmentation on the Pascal VOC dataset. In addition, we evaluate for referring segmentation on the RefCOCO dataset. In summary, we present a first zero-shot, open-vocabulary, unsupervised (no localization information), semantic grounding technique leveraging diffusion-based generative models with no re-training. Our code will be released publicly.

1. Introduction

Image segmentation, a long-standing problem in computer vision, involves partitioning of an image into meaningful spatial regions (or segments) [70]. While semantic segmentation ties the meaning of these segments to a pre-defined set of labels [14,38,89], referring segmentation is more liberal with its open-set labels allowing meaning to be tied to any natural language prompt [26]. The latter also results in multi-modal output spaces; multiple distinct segmentations corresponding to different language prompts can exist for a single given image leading to a more difficult task.

Both tasks are highly useful in numerous real-world applications [17, 26, 69], particularly the latter in human-centric automation [26].

Progress in semantic segmentation utilizing various segmentation-specific deep neural architectures [4,5,62,81,88] reliant on expensive manual annotations [10,38,89] for supervision has recently been superseded by learning under weak supervision [18,36,48], particularly leveraging contrastive image language pre-training models [30,50]. In referring segmentation, the natural language component has driven even early work to utilize language-specific architectural components [26], with recent work [77] similarly building off [50]. While some recent semantic segmentation approaches have been able to eliminate reliance on pixel-wise human annotations for training [53,78,86] operating fully unsupervised for segmentation, these approaches often fail with more complex language prompts, particularly
While contrastive image language pre-training based models [50] have acted as strong foundation models for these segmentation tasks [18, 77], their counterpart in the generative domain - diffusion models [23, 67, 68] - while showcasing impressive performance on realistic text-based image generation [45, 51, 52, 59, 83], have not been utilized for segmentation tasks (to the best of our knowledge). Moreover, most approaches building off diffusion-based foundation models have been limited to generative tasks (e.g. [49]). We ask the question, can pre-trained diffusion models' understanding of separate visual concepts be leveraged to associate natural language to relevant spatial regions of an image? In this work, we explore how stable diffusion models [56] contain such information necessary for the localization of language onto images and how they can act as foundation models for segmentation tasks; in particular, we attempt unsupervised semantic and referring segmentation.

The result, our proposed Peekaboo, is the first unsupervised approach capable of both semantic and referring segmentation. We perform segmentation under zero-shot, open-vocabulary settings with no segmentation-specific architectures or train objectives. Moreover, our approach simply uses an off-the-shelf pre-trained image-language stable diffusion model [56] to perform segmentation with no re-training - i.e. Peekaboo uses the exact same weights as the original model retaining all of its characteristics and strengths. Our contribution is an inference time optimization technique that allows extracting localization-related information contained within these diffusion models.

The proposed inference time optimization involves iteratively updating an alpha mask that converges to the optimal segmentation for a given image and paired language caption. We explore implicit neural representations for improved learning of the alpha mask and proposed a novel alpha compositing-based loss whose optimization generates a suitable segmentation.

The key contributions of this work are as follows:

1. Introducing a novel mechanism for unsupervised segmentation, applicable in both semantic and referring segmentation settings
2. Establishing the presence of pixel-level localization information within pre-trained text-to-image diffusion models
3. Provide a mechanism for utilizing stable diffusion models as off-the-shelf foundation models for downstream segmentation tasks

We evaluate our proposed approach on modified RefCOCO [33] (RefCOCO-C) and Pascal VOC [14] (Pascal VOC-C) datasets to showcase performance in semantic and referring segmentation tasks.

2. Related Work

Vision Language Models: Recently, vision-language models have made rapid advances, expanding into a wide range
of tasks given their impressive generalization abilities [50]. While early work leveraged separate visual and language pre-training for zero-shot image recognition [15, 66] and later language generation from visual inputs [32,34,41,75]. The scaled-up versions of contrastive language image pre-training models [30, 50] have resulted in the recent extensive interest in this domain. Their key significance is the open-vocabulary (ability to operate on free-form language) and zero-shot (no training on downstream tasks) nature. More recent work extends these to a wide variety of tasks [11, 12, 31,47, 79, 82, 85, 87]. Interestingly, these approaches have also been extended for grounding language to images [18, 21] leveraging some localization based supervision (e.g. bounding boxes, masks) but retaining the open-vocabulary characteristics. More recent works explore fully unsupervised segmentation with no localization supervision [53, 78]. While our work contains resemblances to these, our proposed Peekaboo differs through the use of an off-the-shelf diffusion model with no segmentation-specific re-training. Our pre-trained model is also trained for the completely different task of natural image generation and contains no architectural biases or losses geared towards segmentation. Moreover, Peekaboo does not require hand-crafted background prompts and can localize more sophisticated compound phrases or sentences.

**Diffusion Models:** Diffusion Probabilistic Models [67] have recently been adopted successfully for language to vision generative tasks [45, 51, 52, 58–60, 83]. Stable denoising objectives [23, 67, 68], scalable guidance techniques [13, 24], large-scale paired image-text datasets [61], and scalable generative modeling architectures have enabled scaling up these methods to photo-realistic image generation. While diffusion models have also been explored in tasks beyond image generation [6, 25, 35, 49], most rely on large-scale cross-modal datasets for training. Recent work on 3D generation [49] and domain adaptation [19] circumvents this by sampling diffusion models through optimization and is closest to our proposed work. While [49] explores 3D generation with standard diffusion models, our proposed Peekaboo operates with more efficient latent diffusion models and explores a different task of image segmentation. Moreover, we are the first method to explore pre-trained diffusion models in a zero-shot fashion for a cross-modal discriminative task such as segmentation.

**Unsupervised Segmentation** Unsupervised segmentation or grouping is a long-standing task studied in computer vision [40]. This generally involves spatial grouping of semantically related concepts contained within an image. While early approaches prior to deep learning focused on discovering groups of pixels based on known spatially-local affinities [9, 54, 64], recent advances in deep learning based self-supervised learning [3] have revived interest in this task. Building on such self-supervised methods, multiple works explore unsupervised segmentation [8, 16, 22, 29]. These approaches train and operate with no semantic labels, performing grouping of image content at pixel level. However, groups generated by such methods have no alignment to the language modality, i.e. they cannot label the groups into semantically meaningful classes or cannot be probed at varying granularity by language phrases (e.g. human vs face vs eyes). In contrast, our proposed Peekaboo allows grouping aligned to natural language, and the grouping can be conditioned on natural language.

**Referring Segmentation** Referring image segmentation [26, 80, 84] is a challenging task involving two modalities: vision and language. This task is highly applicable in the real world for cases like robotics and human-object interaction [77]. Unlike segmentation tasks involving a fixed set of visual categories, referring segmentation arbitrary natural language expressions. While early approaches [26, 37, 39, 42] fuse features of each modality extracted from deep neural networks to generate segmentation masks, later works explore cross-modal interaction through attention mechanisms [7, 27, 28, 63, 80] while limited to single modality pre-training. Recent [77] is the first to leverage cross-modal pre-training [50] for this task. However, all these methods rely on task-specific re-training requiring human-annotated segmentation masks. To the best of our knowledge, our proposed Peekaboo is the first to perform unsupervised referring segmentation.

### 3. Background

We first present necessary background on diffusion models, latent diffusion, and implicit neural representations.

**Diffusion Models**, a class of generative models, iteratively transform samples (denoising) from a tractable noise distribution towards a specific data distribution (often natural images) [23, 67]. Their conditional variant [45, 51] utilizes an alternate modality, often in the form of language prompts or corrupted images, paired with the specific data distribution. The denoising objective for such a model $Q_θ(x, c)$ is,

$$E_{x,c,t,ε} [w_t \cdot \| Q_θ(α_t x + σ_t ε, c) - ε \|^2_2]$$  \hspace{1cm} (1)

where $(x, c)$ are data-condition pairs, $t \sim U([0, 1])$ is the iteration step, $ε$ is sampled from the noise distribution, $α_t$ and $σ_t$ are functions of $t$ that determine the data-noise ratio, and $w_t$ is an iteration specific weight factor.

**Latent Diffusion** employs a perceptual compression model [56] to project (and revert) the data distribution to a latent space, where the conditional diffusion process operates. We build off an implementation of this, namely, LDM-KL [56]. The key distinction of our proposed Peekaboo is the use of
Can diffusion models separate foreground and background? Stable diffusion was only trained on RGB images. However, observing the often clean boundaries between objects in generated images, it begs the question: can we use these boundaries to generate images with transparency? In this figure, we conduct a self-contained experiment that answers this question: yes. We overlay 6 learnable foreground images on top of 5 learnable background images using 6 learnable alpha masks, to get a total of 30 learnable composite images. Each of these composite images has a composite caption, created by combining the foreground prompt with the background prompt. For example, the image created by combining background #1 and foreground #1 is accompanied by the prompt "a bulldozer by a castle". We optimize each of these 30 composite cells with our proposed dream loss (see Sec. 4.4) until each foreground, background and alpha mask has been learned. Differences between this experiment and Peekaboo are minimal: in Peekaboo, we only optimize the alpha masks, whereas in this analogous experiment we optimize everything.

Implicit Neural Representations, inspired by neural radiance fields (NeRF) [43], generally represent a 3D scene in the form of a neural function [1, 20, 44, 46, 72, 74, 76]. In the 2D domain, raster-based neural textures are similarly used to create realistic images [55, 73]. Focused on modeling surface textures as continuous pixel-less parametric functions, [2] introduces neural-neural textures utilizing Fourier features [71]. Peekaboo builds off this, to implicitly represent alpha-channels in our iterative segmentation process.

4. Proposed Method

In this section, we discuss our proposed approach for unsupervised segmentation, Peekaboo. We leverage a text-to-image stable diffusion model pre-trained on a large internet-scale dataset to segment images in a zero-shot fashion. An alpha-channel compositing process is used to iteratively generate segmentations for the regions of interest in a given image. The zero-shot functionality is endowed by our score distillation mechanism operating in latent space, that connects the language and vision modalities. We represent the alpha-channels used in the iterative segmentation process as implicit neural functions, allowing latent-space score distillation. In the following, we discuss the key components of our proposed system: model architecture, alpha-channel compositing, dream loss, and auxiliary losses. We highlight that our loss functions are only used for inference time optimizations and not for any re-training of diffusion models.

4.1. Peekaboo: Architecture

Given an image \( x_i \), Peekaboo segments the region of interest denoted by a text caption \( c_i \). We utilize a pre-trained diffusion model, \( Q_\theta \), to measure cross-modal similarity. This is used to iteratively generate the expected segmentation mask, \( y_i \). We operate our diffusion process in latent space using a perceptual image compression autoencoder from [56]. The encoder, \( E \), down-samples some input \( \hat{x}_i \in R^{H \times W \times 3} \) by a factor of \( f = 8 \) to produce a latent \( z_i \in R^{H/f \times W/f \times 4} \) while the decoder, \( D \), reverts \( z_i \) to produce \( \hat{x}_i \). Text captions are encoded using a CLIP ViT-L/14 [50] pre-trained text encoder, \( T \), and the output prior to the final average pooling operation is used as the feature vector. The diffusion denoising process is performed by the variant of U-Net [57] that is used in [56]. This is a modification of the standard diffusion model U-Net from [13], incorporating a position-wise multi-layer perceptron and a cross-attention operation (for conditioning on the encoded text caption). The final segmentation prediction of our system, \( \hat{y}_i \), represented as a learnable implicit neural function \( \psi_{\phi_i} \), is a single-channel alpha mask. We generate multi-
4.2. Learnable Alpha Compositing

Alpha compositing or alpha blending, a standard tool in all computer graphics, involves combining a given image with a given background, creating an appearance of partial or full transparency [65]. In this work, we introduce the concept of a learnable alpha mask that focuses on a region of interest in a given image. While alpha masks (or any image like data) are generally represented as array data structures (matrices), here we use an implicit neural representation, for reasons detailed in Sec. 4.4.

Similar to our analogous experiment (Fig. 3), the learnable alpha mask is used to composite the input image, \( x_i \), with a background, \( b_j \). In this case, we use random colored uniform backgrounds (all pixels of the same color). This results in the composite image, \( \hat{x}_{i,j} \), being a masked version of the original image (only a limited spatial region visible) with mask color from \( b_j \). We generate a set of \( n_b \) outputs with multiple random colored backgrounds using the single alpha mask. The region visible across all composite images is dictated by the learnable alpha mask.

Our proposed dream loss (Sec. 4.4), \( L_d(\hat{x}, c) \), can be viewed as measuring the cross-modal similarity between the composite image and text caption. We iteratively optimize the alpha mask, \( \hat{y} \), with respect to our dream loss, enforcing each visible region to relate to its common text prompt. This optimization converges with the alpha mask localizing to the region relevant to the input text caption.

Since a single image may contain multiple regions relevant to different text captions (\( c_{i,k} \)), we learn separate alpha masks (\( \hat{y}_{i,k} \)) for each text caption. Optimizing each of these in parallel allows for segmenting each region separately.

\[
\hat{x}_{i,j,k} = \hat{y}_{i,k} \cdot x_i + (1 - \hat{y}_{i,k}) \cdot b_j \tag{2}
\]

\[
L_{cd}(x_i, c_i) = \sum_{k=1}^{n_b} \sum_{j=1}^{H \times W} L_d(\hat{x}_{i,j,k}, c_{i,k}) \tag{3}
\]

Therein, for a single image \( x_i \) containing multiple captions, \( c_{i,k} \), we obtain a compound dream loss, \( L_{cd} \).

4.3. Implicit Mask Representation

The learnable alpha mask, \( \hat{y}_i \), is represented as a function modeled by a neural network. In particular, a parametric Fourier domain representation is used, inspired by [2]. We utilize a multi-layer perceptron (MLP) network, \( \psi_{\phi_i} \), with 4 fully-connected layers to generate the mask parameters. The input to the MLP network is a set of Fourier domain parameters \( u, v \) while the output is the set of \( k \) alpha masks relevant to each per image caption, i.e. \( \hat{y}_i \in \mathbb{R}^{k \times H \times W} \).

\[
\psi_{\phi_i} : (u, v) \in \mathbb{R}^2 \rightarrow (\alpha_1, \alpha_2, ... ) \in \mathbb{R}^k \tag{4}
\]

\[
\hat{y}_i = \psi(u, v; \phi_i) \tag{5}
\]

Given this implicit mask representation, for each image, \( x_i \), to be segmented during inference, we iteratively update \( \phi_i \),
the parameters defining the mask $\hat{y}_t$. In terms of this parameterization, we note that operating in the Fourier domain easily supports meaningful constraints on the mask. Furthermore, the implicit mask representation allows,

1. predicting a modality (alpha masks) different to train time data (RGB)
2. applying constraints in the form of various loss functions to the optimization
3. generating multiple masks (pertaining to the different regions of interest within an image) from a common network

We note how the latter allows modeling the relations across the entire image during the mask generation process.

4.4. Dream Loss

We now explain our proposed Dream Loss that connects the text and image modalities to generate the expected segmentation mask. Our latent diffusion model (LDM) from [56] pre-trained on LAION-5B [61] (and subsets) contains extensive cross-modal information. We utilize this LDM to measure the cross-modal similarity between visual and language domains. Despite the term "loss", we reiterate that it is not used for any training whatsoever.

Given an image-text pair $(\hat{x}, c)$, the diffusion model, $Q_\theta$, estimates the noise content, $\hat{\epsilon}$, necessary to iteratively update the initial noise vector, $\epsilon$, towards the final output to be generated at each time-step. In essence, it estimates the noise given a noisy version of the input, $\sigma_t x + \sigma_t \epsilon$ (refer Eq. (1)) together with the text caption, $c$. In our case, this process operates within a latent space. Given this setup, we reformulate Eq. (1) as Eq. (6),

$$\mathbb{E}_{x, c, \epsilon, t} \left[ w_t \cdot \left\| Q_\theta(\mathcal{E}(\hat{x}[t]), \mathcal{T}(c), t) - \epsilon \right\|^2 \right]$$

where the addition $t$ is the time step. While this objective alone can serve as a cross-modal similarity measure, gradient updates through the diffusion model, $Q_\theta$, can be computationally expensive and lead to optimization instability. Inspired by [19,49], we build off this base objective in Eq. (6) and ignore gradients across the diffusion model, to obtain our proposed dream loss, $L_d$ as follows:

$$\hat{\epsilon}_t = Q_\theta(\mathcal{E}(\hat{x}[t]), \mathcal{T}(c), t)$$

$$\nabla_\phi L_d(\hat{x}, c) = \mathbb{E}_{\epsilon, t} \left[ w_t \cdot (\hat{\epsilon}_t - \epsilon) \cdot \frac{\partial \hat{y}}{\partial \phi} \right]$$

with Eq. (2) and Eq. (5) relating composite image, $\hat{x}$, to learnable mask, $\hat{y}$, and mask parameters, $\phi$.

Therein, we optimize our alpha mask, $\hat{x}$ with respect to our dream loss, $L_d$, in order to obtain the optimal segmentation conditioned on the text caption, $c$.

4.5. Auxiliary Losses

We include two auxiliary losses to enhance the segmentation process, namely, Gravity Loss ($L_g$) and Intersection Loss ($L_i$). These incorporate heuristics regarding image backgrounds and object intersections respectively. In particular, the Gravity loss assumes the presence of background (or regions unrelated to text caption) outside of the region most similar to the text caption and aims to push the alpha mask values of these regions to zero. This is implemented as a loss on the sum of all alpha mask values, pushing them to zero by default.

$$L_g(\hat{y}) = \sum \hat{y}$$

$$L_f = L_{cd} + L_g + L_i$$

The Intersection loss penalizes masks containing spatially overlapping regions, acting on the heuristic that a single pixel location corresponds to a single object. This second loss is only incorporated in object-based segmentation tasks, and not for referring segmentation where such heuristics may not be applicable. Our total loss, $L_f$ is the summation of all three as in Eq. (10).
Table 1. Results on modified RefCOCO (RefCOCO-C) dataset: we report mIoU values for our method on a modified version of this dataset. Our proposed approach obtains improved performance over most comparisons. Note that DINO [3] obtains a segmentation of the foreground region of each image without using any text caption.

5. Analogous Example

In order to explain the intuition behind Peekaboo, we show results of analogous experiment that helped to inspire it.

We first examine a stable diffusion model [56] pre-trained on LAION-5B [61]. Our goal is to explore whether internal knowledge of these models regarding boundaries and localization of individual objects can be accessed and subsequently utilized for tasks such as segmentation. We focus on the case of generating a single synthetic object in some background and attempt to generate an accompanying alpha mask that demarcates the region belonging to the foreground object. We utilize our proposed dream loss (see Sec. 4.4) as a cross-modal similarity function that connects a text caption describing a foreground object to the image region it is located. Using this similarity as an optimization objective, we generate the foreground, background, and alpha mask. We use implicit neural functions (see Sec. 4.3) to represent each of these elements, with one network for background and a shared network for foreground and alpha mask. The latter design choice is aimed at sharing information between the generated foreground and alpha mask.

Results obtained from this process are illustrated in Fig. 3. While our method is able to generate good segmentation masks relevant to the foreground, we note that generated images are unrealistic. We highlight that generated image quality is indifferent to the segmentation component, where we generate single-channel alpha masks. The rest of our work is focused on how this technique can be leveraged for segmenting stand-alone images, i.e. images beyond those generated by a diffusion model. In essence, we attempt to segment real-world images with free-form text captions.

6. Experiments

In this section, we detail our experimental setup used to evaluate Peekaboo and present results, both quantitative and qualitative.

Tasks & Datasets: We evaluate Peekaboo on two tasks, semantic segmentation and referring segmentation, selecting Pascal VOC [14] and RefCOCO [33] datasets to experiment on for each task, respectively. Pascal VOC is a semantic segmentation dataset containing objects belonging to 20 different classes and a background class. It contains 1450 images in its validation set. RefCOCO is one of the largest and most commonly used datasets for referring image segmentation. It contains 10,834 validation, 5,657 test A, and 5,095 test B samples, respectively. The images are collected from the MS-COCO [38] via a two-player game [33].
Figure 7. **Images generated from Stable Diffusion**: This data distribution is quite different from the distribution of natural images, rarely containing objects in non-central locations of the image. These images were generated by running prompts through stable diffusion, such as “a teddy bear in times square”

Being the first generative diffusion model-based segmentation approach, the strengths of our model are unique. Therein we apply some pre-processing to the datasets that highlight key strengths of our model. All details of dataset modifications are outlined in the appendix. The reason Peekaboo works better with our pre-processed images is that Peekaboo is best at segmenting images that look similar to the outputs of Stable Diffusion. For example, if we use the prompt "a teddy bear in times square", we are unlikely to get a small teddy bear hiding in the top left corner of the image (see Fig. 7). We are much more likely to obtain a large teddy bear in the foreground. Since peekaboo is based on this diffusion model, it will therefore be biased toward finding large teddy bears in the middle of the screen.

Training Details: Our approach requires no re-training of the model and only involves inference time optimization. This optimization process is run for 200 iterations. A drawback of our method is the high computational cost during inference. Segmenting a single image takes approx—3 minutes on a single NVIDIA RTX 3090Ti GPU.

### 6.1. Referring Segmentation

We present our quantitative evaluations for referring segmentation in Tab. 1. Since there are no reported unsupervised methods for comparison, we first build a baseline that predicts the entire image as the segmentation (inspired by [26]) which we tag as "random (whole image)". We build a second baseline using DINO [3]. For a given image, DINO is able to generate segmentations of the foreground. Since these are not conditioned on the paired text caption, this is comparable to a random foreground segmentation of the image. We next apply the pre-trained model from GroupViT [78] to obtain segmentations conditioned on the text captions. We highlight that GroupViT is intended to be used for semantic segmentation tasks, and here we apply it in an out-of-domain referring segmentation task here, mainly for comparison purposes. Our results presented in Tab. 1 showcase how we surpass other baselines on the referring segmentation task. We reiterate that our method operates unsupervised performing no re-training of the diffusion model for the downstream task. We additionally present a per-class distribution of mIoU values for our method in Fig. 6.

| Method    | Foreground mIoU |
|-----------|-----------------|
| GroupViT [78] | 0.20            |
| Ours      | 0.35            |

Table 2. Results on modified Pascal VOC dataset: we report mIoU values for our method on a modified version of this dataset. Numbers for GroupViT are obtained from running their pre-trained model on our modified version of the dataset.

### 6.2. Semantic Segmentation

We present evaluations on the modified Pascal VOC dataset in Tab. 2. We compare against the baseline of GroupVit, which is another unsupervised approach for semantic segmentation. We note that reported results are for our modified version of Pascal VOC. Per-class mIoU values are also reported in Fig. 6.

**7. Conclusion**

In this work, we established how text-to-image diffusion models trained on large-scale internet data contain strong cues on localization. Using an analogous experiment Fig. 3, we demonstrated how an image generation process can be augmented to concurrently generate a segmentation for the foreground object. Motivated by this behavior, with a small tweak we introduced a novel inference time optimization technique for utilizing off-the-shelf text-to-image diffusion models in segmentation tasks. Our approach can be applied to downstream segmentation tasks with no segmentation-specific re-training of the diffusion model.

**Reproducibility & Ethics**

We utilize publicly available pre-trained stable diffusion models for all our experiments. All code relevant to our contribution will be released publicly. We do acknowledge how pre-trained diffusion models we use contain harmful biases which will be transferred to segmentation by our proposed method.
References

[1] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. CVPR, 2022.

[2] Ryan Burgert, Jinghuan Shang, Xiang Li, and Michael Ryoo. Neural network textures make sim2real consistent. In Proceedings of the 6th Conference on Robot Learning, 2022.

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégo, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In CVPR, pages 9650–9660, 2021.

[4] Liang-Chieh Chen, Maxwell Collins, Yukun Zhu, George Papandreou, Barret Zoph, Florian Schroff, Hartwig Adam, and Jonathon Shlens. Searching for efficient multi-scale architectures for dense image prediction. NeurIPS, 31, 2018.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE TPAMI, 2017.

[6] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. arXiv:2009.00713, 2020.

[7] Yi Wen Chen, Yi Hsuan Tsai, Tiantian Wang, Yen Yu Lin, Nanxin Chen, Yu Zhang, Heiga Zen, Ron J. Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. arXiv:2009.00713, 2020.

[8] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Minh Hsuan Yang. Referring expression object segmentation with caption-aware consistency. In BMVC. 2019.

[9] Jang Hyun Cho, Utkarsh Mall, Kavita Bala, and Bharath Hariharan. Picie: Unsupervised semantic segmentation using invariance and equivariance in clustering. CVPR, pages 16789–16799, 2021.

[10] Dorin Comaniciu and Peter Meer. Robust analysis of feature spaces: Color image segmentation. In CVPR, pages 750–755. IEEE, 1997.

[11] 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. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3213–3223, 2016.

[12] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. In CVPR, pages 11162–11173, 2021.

[13] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. ArXiv, abs/2105.05233, 2021.

[14] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. IJCV, 2010.

[15] Andrea Frome, Greg S Corrado, Jonathon Shlens, Samy Bengio, Jeff Dean, Marc’Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. NeurIPS, 26, 2013.

[16] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Unsupervised semantic segmentation by contrasting object mask proposals. ICCV, pages 10032–10042, 2021.

[17] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In CVPR, pages 3354–3361. IEEE, 2012.

[18] Golnaz Ghiassi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Open-vocabulary image segmentation. ArXiv, abs/2112.12143, 2021.

[19] Alexandros Graikos, Nikolay Malkin, Nebojsa Jojic, and Dimitris Samaras. Diffusion models as plug-and-play priors. In Thirty-Sixth Conference on Neural Information Processing Systems, 2022.

[20] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. StyleNeRF: A style-based 3d-aware generator for high-resolution image synthesis, 2021.

[21] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In ICLR, 2022.

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

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. NeurIPS, 2020.

[24] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. arXiv:2207.12598, 2022.

[25] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. arXiv:2204.03458, 2022.

[26] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In ECCV, 2016.

[27] Shaofei Huang, Tianrui Hui, Si Liu, Guanbin Li, Yunchao Wei, Jizhong Han, Luojiu Liu, and Bo Li. Referring image segmentation via cross-modal progressive comprehension. In CVPR, 2020.

[28] Tianrui Hui, Si Liu, Shaofei Huang, Guanbin Li, Sansi Yu, Faxi Zhang, and Jizhong Han. Linguistic structure guided context modeling for referring image segmentation. In ECCV, 2020.

[29] Xu Ji, Andrea Vedaldi, and João F. Henriques. Invariant information clustering for unsupervised image classification and segmentation. ICCV, pages 9864–9873, 2019.

[30] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. Int. Conf. on Mach. Learn., 2021.

[31] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdeit-modulated detection for end-to-end multi-modal understanding. In ICCV, pages 1780–1790, 2021.

[32] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In CVPR, pages 3128–3137, 2015.
A. Intuition

We explore the reasoning behind each component used and explain why Peekaboo works as it does. The key underlying idea is deriving gradients from a conditional diffusion model to guide a mask generation process.

The diffusion model conditioned on a text caption processes a noisy input to generate a gradient that moves the input toward the target image. This gradient is what we use to guide our learnable alpha mask. We use Stable Diffusion [56], a text-to-image diffusion model trained on billions of internet images. This means the model will target to generate images from that image distribution relevant to a text caption. The noisy input is the image to be segmented alpha blended with a uniform background. Iteratively, it will attempt to update this input to a target distribution image relevant to the text caption. Updating the regions of the image most relevant to the text caption makes more sense. Thus, regions of the image relevant to this text caption will have stronger gradients than regions irrelevant to a prompt. For example, consider an image of a dog sitting on a couch. When prompted for “a dog”, the couch is irrelevant to the prompt and has fewer gradients focused on that region. There is less incentive to remove the alpha mask in that location, and without incentive, it defaults to null due to our gravity loss (Sec. 4.5). This results in the alpha mask focusing on the dog region.

While this results in our desirable behavior, we observe that it converges to the region most relevant to the text prompt. For example in Fig. 9, in segmenting Emma Watson, it attempts to focus on the region that mostly makes the image look like her, which could be a sub-portion of the human region (in accordance with the accepted notion of segmenting a human). While Harry Styles could wear a dress, he would still remain Harry Styles. However, only Emma Watson can have her face. Therefore, Emma Watson’s face is more essential to the prompt Emma Watson; hence it prioritized segmenting that region while ignoring the dress. In a way, one could view this as a new definition of segmentation; Peekaboo localizes the essence of an object described by language.

B. Ablations

In this section, we present some ablations on components of our inference time optimization.

B.1. Implicit neural representations

In Peekaboo, we use the neural-neural texture formulation from [2] to represent our learnable alpha masks (and foreground/background in our analogous example in ??). The alternative to using this implicit neural representation is learning the pixels directly by representing them as a learnable tensor of dimensions \((H, W, C)\) for \(C = 1\) in mask and \(C = 3\) in RGB images. In this ablation, we explore how the alternative compares against our selected approach. We highlight the two main drawbacks of pixel-level representations as 1) noisy outputs and 2) bad convergence.

We illustrate this behavior in Fig. 11. First, we attempt to
Figure 10. **Peekaboo is truly open vocabulary**: We illustrate examples where Peekaboo is able to localize various regions of interest defined by references from popular culture. All of these are examples where the model has been able to correctly localize (identified region centroid or high IoU with correct region). The caption below each image is used to generate the same color mask. An interesting behavior of our model is its localization to the region most defined by the accompanying text caption. For example, in the case of Emma Watson in the top row second figure, it is visible how Peekaboo localizes on her face and body instead of her frock (see Fig. 9 for more on this).

Figure 11. **Ablation on implicit neural representation**: In Peekaboo we utilize neural-neural textures [2]. Here we compare against the alternative of direct pixel-level representations. (top) The first row shows images generated using implicit representations while the second row shows those of pixel-level ones. The two graphs below correspond to pixel variance (y-axis) plotted against iteration (x-axis) for implicit and pixel-level representation respectively from left to right.

generate images (similar to ??) using each of the two methods and show that neural representations lead to less noisy outputs. The top two rows in Fig. 11 correspond to these. Clearly, the neural representation in the first row leads to a less noisy output in comparison to the alternative. Next, we explore the convergence behavior of each approach. To analyze this, we utilize the images being generated and measure their variance at the pixel level. While a natural image contains some variance in pixel space, this is a finite value, and our run-time optimization should ideally converge at this variance. However, utilizing a pixel-level representation results in continuously increasing variance in the image pixels, leading to overly saturated images (dissimilar to a realistic image) and in the case of masks, lack of convergence. This is illustrated in the two graphs at the bottom of Fig. 11. We also highlight how in these graphs of Fig. 11 the variance of the implicit representation converges early on at around 50 iterations (left) while that of the pixel-wise
B.2. Bilateral Filter

Another component of our proposed approach is a bilateral filter conditioned on the image to be segmented. This filter is applied onto the learnable alpha mask that is represented implicitly. In detail, the rasterized output of the implicit representation (pixel-wise mask) is modified using the bilateral filter. A closer look at the noisy mask in row 3 column 1 of Fig. 13 will show vague outlines of a skeleton within the noise; this is a result of the bilateral filter being applied on the mask.

The motivation for using such a filter is to align our generated segmentation masks with the image content at a pixel level, i.e. to respect the boundaries between regions present in the image. This bilateral filter results in the mask being aligned to these from early iterations, leading to a better segmentation at the end. This behavior is illustrated in Fig. 13.

C. Optimization Details

In this section, we discuss all details relevant to our proposed inference time optimization that generates segmentations. In order to generate a single segmentation, we run it for 200 iterations using a batch size of 1, a learning rate of $10^{-5}$, and stochastic gradient descent as the optimizer.

The implicit neural representation follows the neural-neural textures from [2], using the same number of layers for the MLP, the same Fourier features, and the same scale.

Additionally, we apply a bilateral blur operation, conditioned on the image to be segmented, onto the learnable alpha masks at initialization, which results in faster and better convergence. Here we use a blur kernel of size 3 with 40 iterations (multiple iterations increase the effective field of view for a kernel).

Figure 13. Ablation on bilateral filters: In this figure, we show a timelapse of the alpha mask optimization process over time from left to right. We utilize a bilateral filter conditioned on the image to initialize the learnable alpha mask. This results in better segmentations that are more aligned to the boundaries of the objects contained within the image. The rows from top to bottom show, a) overlay with bilateral filter, b) overlay without bilateral filter, c) mask with bilateral filter, and d) mask without bilateral filter. We highlight how the bilateral filter leads to better alignment of segmentation to the object boundaries.

D. Visualizations

In this section, we present some more qualitative evaluations of our proposed method. In Fig. 10, we show some randomly selected examples where Peekaboo successfully localizes regions of interest based on popular cultural references. The stable diffusion model used in our dream loss is pre-trained on a large corpus of internet image-text pairs.
Figure 14. **Limitations and failures of Peekaboo.** We illustrate examples where Peekaboo fails to properly segment the region of interest. The prompts for these examples from left to right are: knife, eyes, bird, cat, dog. We note how a major issue is the tendency of Peekaboo to hallucinate the shape of the object denoted by the text caption.

We hypothesize that our Peekaboo is able to understand such a wide vocabulary due to the strength of the pre-trained diffusion model. We showcase another example of a real-world image captured by our camera in Fig. 8.

In comparison to existing unsupervised open vocabulary localization methods (particularly ones trained in a discriminative fashion), we highlight that Peekaboo is truly open vocabulary covering a wide range of concepts that can be encoded in natural language. The generative objectives used for the diffusion model pre-training necessitate learning the image-text distribution it is trained on (as opposed to boundaries). We hypothesize that this generative objective provides the model with a holistic and extensive understanding of that data distribution. It is this understanding that Peekaboo leverages and extracts using its inference time optimization to perform such open vocabulary unsupervised segmentation.

Another characteristic of this open vocabulary nature that endows our model is to probe image regions at varying granularities. We illustrate this behavior in Fig. 9 where sub-regions of a single person can be localized based on different text captions.

More results on RefCOCO-C and Pascal VOC-C are shown as Fig. 18 and Fig. 16.

**E. Limitations**

We also acknowledge that our proposed Peekaboo contains various limitations and shortcomings. A key drawback is its failure cases that result in a hallucination of the text prompt using some random background region, i.e. it uses the background texture to create a region shaped like the underlying object described by the text. We illustrate this behavior in Fig. 14. This is clearly visible in the three right-most examples (bird, cat, dog). We also note that such behavior is more common when a simple (often one word) text caption is used. Another failure is the addition of unnecessary parts to the region of interest. For example, in column one of Fig. 14, while the knife is coarsely localized, Peekaboo also incorrectly creates a handle for it using a slice of bread from the background. The model also sometimes fails to converge entirely, as illustrated in the second column, where despite localizing the eyes, the generated mask also holds onto outlines of the image foreground object.

While we hope to address these issues in future work, we also reiterate that despite these limitations, Peekaboo is a first unsupervised method that is able to perform open vocabulary segmentation using arbitrary natural language prompts.
Figure 15. Our results on RefCOCO-C. For each sample we demonstrate the prompt text (title), input image (left), our output alpha mask (middle), and the image segmented by the mask (right).
Figure 16. Our results on Pascal VOC-C. For each sample we demonstrate the prompt text (title), input image (left), our output alpha mask (middle), and the image segmented by the mask (right).
Figure 17. Extended visualization of analogous experiment
Figure 18. In this figure, we play a game: find the eyes! Peekaboo can segment specific features of images, and is good at finding eyes. Only one prompt was used: eyes. On the top row we have input images, and in the middle row we have the outputted alpha map. On the bottom we overlay the input images with a white mask corresponding to the alpha, to better show which part of the image it chose to segment.