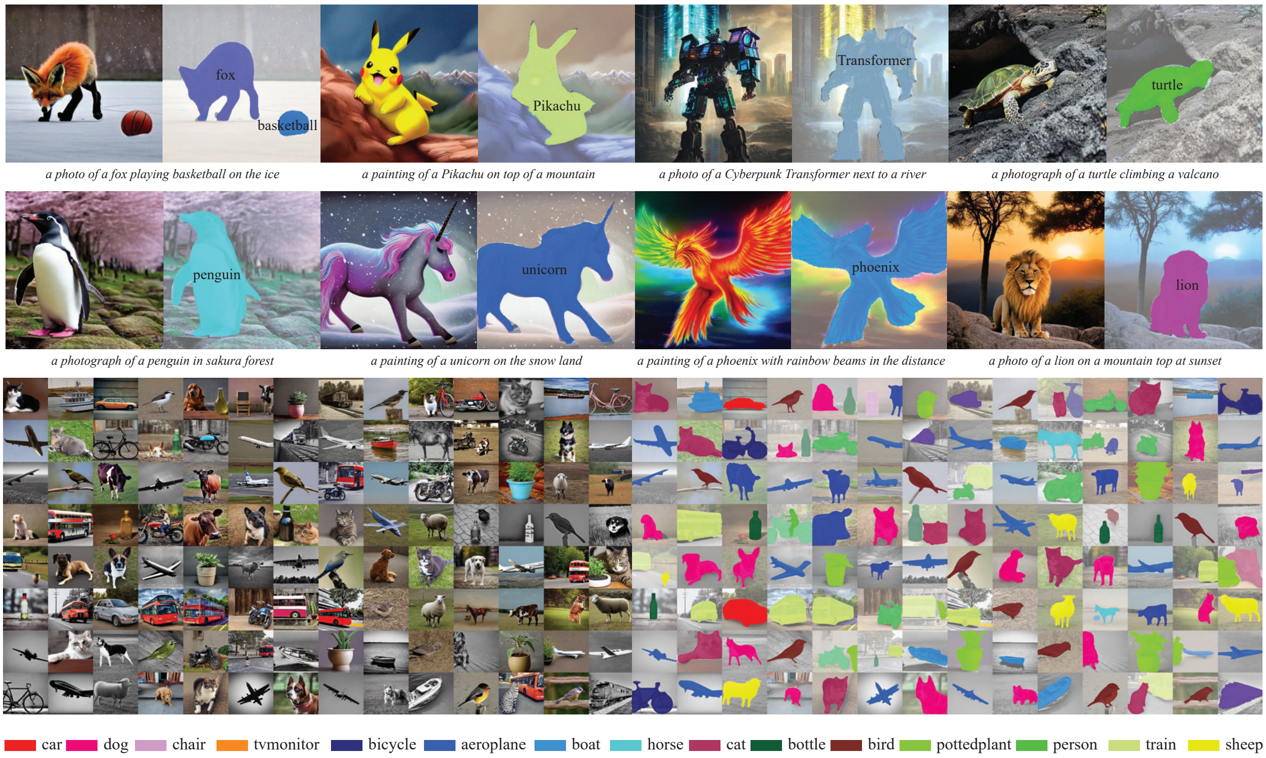
Open-vocabulary Object Segmentation with Diffusion Models

Ziyi Li*[1], Qinye Zhou*[1], Xiaoyun Zhang[1], Ya Zhang[1,2], Yanfeng Wang†[1,2], and Weidi Xie†[1,2]
[1]Coop. Medianet Innovation Center, Shanghai Jiao Tong University, China
[2]Shanghai AI Laboratory, China
{599lzy, zhouqinye, xiaoyun.zhang, ya_zhang, wangyanfeng, weidi}@sjtu.edu.cn
https://lipurple.github.io/Grounded_Diffusion/



a photo of a fox playing basketball on the ice

a painting of a Pikachu on top of a mountain

a photo of a Cyberpunk Transformer next to a river

a photograph of a turtle climbing a valcano

a photograph of a penguin in sakura forest

a painting of a unicorn on the snow land

a painting of a phoenix with rainbow beams in the distance

a photo of a lion on a mountain top at sunset

Figure 1: **Predictions from our guided text-to-image diffusion model**. The model is able to simultaneously generate images and segmentation masks for the corresponding visual objects described in the text prompt, for example, Pikachu, Unicorn, Phoenix, *etc*.

Abstract

The goal of this paper is to extract the visual-language correspondence from a pre-trained text-to-image diffusion model, in the form of segmentation map, i.e., *simultaneously generating images and segmentation masks for the corresponding visual entities described in the text prompt. We make the following contributions: (i) we pair the existing Stable Diffusion model with a novel grounding module, that can be trained to align the visual and textual embedding space of the diffusion model with only a small number of object categories; (ii) we establish an automatic pipeline for constructing a dataset, that consists of {image, segmentation mask, text prompt} triplets, to train the proposed grounding module; (iii) we evaluate the performance of open-vocabulary grounding on images generated from the text-to-image diffusion model and show that the module can well segment the objects of categories beyond seen ones at training time, as shown in Fig. 1; (iv) we adopt the augmented diffusion model to build a synthetic semantic segmentation dataset, and show that, training a standard segmentation model on such dataset demonstrates competitive performance on the zero-shot segmentation (ZS3) benchmark, which opens up new opportunities for adopting the*

* Both the authors have contributed equally to this project.
† denotes corresponding author.

powerful diffusion model for discriminative tasks.

1. Introduction

In the recent literature, text-to-image generative models have gained increasing attention from the research community and wide public, one of the main advantages of such models is the strong correspondence between visual pixels and language, learned from large corpus of image-caption pairs, such correspondence, for instance, enables to generate photorealistic images from the free-form text prompt [26, 29, 39, 25]. In this paper, we aim to explicitly extract such visual-language correspondence from the generative model in the form of segmentation map, *i.e.*, simultaneously generating photorealistic images, and infer segmentation masks for corresponding visual objects described in the text prompts. The benefit of extracting such visual-language correspondence from generative model is significant, as it enables to synthesize infinite number of images with pixel-wise segmentation for categories within the vocabulary of generative model, serving as a free source for augmenting the ability of discriminative segmentation or detection models, *i.e.*, be able to process more categories.

To achieve such goal, we propose to pair the existing Stable Diffusion [26] with a novel grounding module, that can segment the corresponding visual objects described in the text prompt, from the generated image. This is achieved by explicitly aligning the text embedding space of desired entity and the visual features of synthetic images, *i.e.*, intermediate layers of diffusion model. Once trained, objects of interest can be segmented by their category names, for both seen and unseen objects at training time, resembling an open-vocabulary object segmentation for generative model.

To properly train the proposed architecture, we establish a pipeline for automatically constructing a dataset with $\{$synthetic image, segmentation mask, text prompt$\}$ triplets, in particular, we adopt an off-the-shelf object detector, and do inference on images generated from the existing Stable Diffusion model, advocating no extra manual annotation. Theoretically, such a pipeline enables to generate infinite data samples for each category within the vocabulary of existing object detector, for example, we adopt the Mask R-CNN [21] pre-trained on COCO with 80 categories. We show that the grounding module trained on a pre-defined set of object categories, can segment images from Stable Diffusion well beyond the vocabulary of any off-the-shelf detector, as shown in Fig. 1, for example, Pikachu, unicorn, phoenix, *etc*, effectively resembling a form of visual instruction tuning, for establishing visual-language correspondence from generative model.

To quantitatively validate the effectiveness of our proposed grounding module, we initiate two protocols for evaluation: *first*, we compare the segmentation results with a strong off-the-shelf object detector on synthetic images; *second*, we construct a synthesized semantic segmentation dataset with Stable Diffusion and our grounding module, then train a segmentation model on it. While evaluating zero-shot segmentation (ZS3) on COCO and PASCAL VOC, we outperform prior state-of-the-art models on unseen categories and show competitive performance on seen categories, reflecting the effectiveness of our constructed datasets. Even more crucially, we demonstrate an appealing application for training discriminative models with synthetic data from generative model, for example, to expand the vocabulary beyond any existing detector can do.

2. Preliminary on Diffusion Model

Diffusion models refer to a series of probabilistic generative models, that are trained to learn a data distribution by gradually denoising the randomly sampled Gaussian noises. Theoretically, the procedure refers to learning the reverse process of a fixed Markov Chain of length T. As for text-to-image synthesis, given a dataset of image-caption pairs, *i.e.*, $\mathcal{D}_{\text{train}} = \{(\mathcal{I}_1, y_1), \ldots, (\mathcal{I}_N, y_N)\}$, the models can be interpreted as an equally weighted sequence of conditional denoising neural network that iteratively predicts a denoised variant of the input conditioned on the text prompt, namely $\epsilon_\theta(\mathcal{I}_i^t, t, y_i)$, where $\mathcal{I}_i^t$ denotes a noisy version of the input image, and $t = 1, \ldots, T$ refers to the timestep, $i \in \{1, \ldots, N\}$. For simplicity, we only describe the training and inference procedure for a single image, thus ignoring the subscript in the following sections.

In particular, this paper builds on a variant of diffusion model, namely, Stable Diffusion [26], which conducts the diffusion process in latent space. We will briefly describe its architecture and training procedure in the following.

Architecture. Stable Diffusion consists of three components: a text encoder for producing text embeddings; a pre-trained variational auto-encoder that encodes and decodes latent vectors for images; and a time-conditional U-Net ($\epsilon_\theta(\cdot)$) for gradually denoising the latent vectors, with the progressive convolutional operation that downsamples and upsamples the visual feature maps with skip connections. The visual-language interaction occurs in the U-Net via cross-attention layers, specifically, a text encoder is used to project the text prompt y to textual embeddings, that then are mapped into `Key` and `Value`, and the spatial feature map of the noisy image is linearly projected into `Query`, iteratively attending the conditioned text prompt for update.

Training and Inference. The training procedure of Stable Diffusion can be described as follows: given a training pair $(\mathcal{I}, y)$, the input image $\mathcal{I}$ is first mapped to a latent vector z and get a variably-noised vector $z^t := \alpha^t z + \sigma^t \epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$ is a noise term and α^t, σ^t are terms that control the noise schedule and sample quality. At training time,

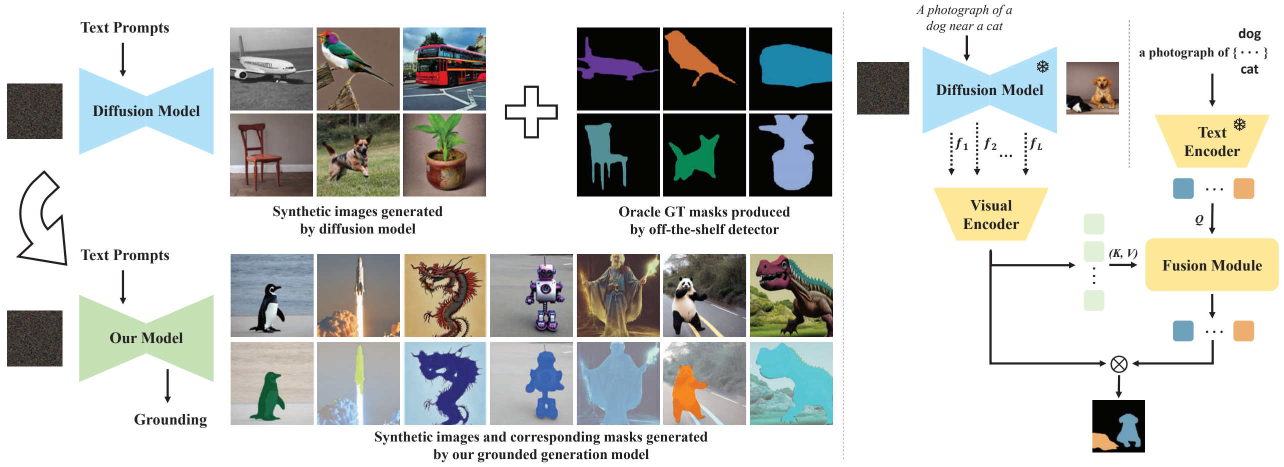

Text Prompts
Diffusion Model
Synthetic images generated by diffusion model
Oracle GT masks produced by off-the-shelf detector
Text Prompts
Our Model
Grounding
Synthetic images and corresponding masks generated by our grounded generation model
A photograph of a dog near a cat
Diffusion Model
f_1 f_2 ... f_L
Visual Encoder
dog
a photograph of { · · · }
cat
Text Encoder
Q
(K, V)
Fusion Module

Figure 2: **Overview.** The **left** figure shows the knowledge induction procedure, where we first construct a dataset with synthetic images from diffusion model and generate corresponding oracle groundtruth masks by an off-the-shelf object detector, which is used to train the open-vocabulary grounding module. The **right** figure shows the architectural detail of our grounding module, that takes the text embeddings of corresponding entities and the visual features extracted from diffusion model as input, and outputs the corresponding segmentation masks. During training, both the diffusion model and text encoder are kept *frozen*.

the time-conditional U-Net is optimised to predict the noise ϵ and recover the initial z, conditioned on the text prompt y, the model is trained with a squared error loss on the predicted noise term as follows:

$$\mathcal{L}_{\text{diffusion}} = \mathbb{E}_{z,\epsilon\sim\mathcal{N}(0,1),t,y}\left[||\epsilon - \epsilon_\theta(z^t, t, y)||_2^2\right] \quad (1)$$

where t is uniformly sampled from $\{1, \ldots, T\}$.

At inference time, Stable Diffusion is sampled by iteratively denoising $z^T \sim \mathcal{N}(0, I)$ conditioned on the text prompt y. Specifically, at each denoising step $t = 1, \ldots, T$, z^{t-1} is obtained from both z^t and the predicted noise term of U-Net whose input is z^t and text prompt y. After the final denoising step, z^0 will be mapped back to yield the generated image $\mathcal{I}$.

3. Problem Formulation

In this paper, we aim to augment an existing text-to-image diffusion model with the ability of open-vocabulary segmentation, by exploiting the visual-language correspondence, *i.e.*, simultaneously generating images, and the segmentation masks of corresponding objects described in the text prompt, as shown in Fig. 2 (left):

$$\{\mathcal{I}, m\} = \Phi_{\text{diffusion}^+}(\epsilon, y) \quad (2)$$

where $\Phi_{\text{diffusion}^+}(\cdot)$ refers to a pre-trained text-to-image diffusion model with a grounding module appended, it takes the sampled noise ($\epsilon \sim \mathcal{N}(0, I)$) and language description y as input, and generates an image ($\mathcal{I} \in \mathbb{R}^{H\times W\times 3}$) with corresponding segmentation masks ($m \in \{0, 1\}^{H\times W\times \mathcal{O}}$) for a total of $\mathcal{O}$ objects of interest. **Note that**, we expect the model to be *open-vocabulary*, that means, it should be able to output the corresponding segmentation mask for any objects that can be generated by diffusion model, without limitation of the semantic categories.

4. Open-vocabulary Grounding

Assuming there exists a training set of N triplets, *i.e.*, $\mathcal{D}_{\text{train}} = \{(\mathcal{F}_1, m_1^{\text{gt}}, y_1), \ldots, (\mathcal{F}_N, m_N^{\text{gt}}, y_N)\}$, the predicted segmentation mask for all objects, *i.e.*, $m_i \in \mathbb{R}^{H\times W\times \mathcal{O}_i}$ can be obtained by:

$$m_i = \Phi_{\text{fuse}}(\Phi_{\text{v-enc}}(f_i^1, \ldots, f_i^n), \Phi_{\text{t-enc}}(g(y_i))) \quad (3)$$

where y_i denotes the text prompt for image generation, $\mathcal{F}_i = \{f_i^1, \ldots, f_i^n\}$ refers to the intermediate representation from Stable Diffusion at $t = 5$ (this has been experimentally validated in supplementary material). Our proposed **grounding module** consists of three functions, namely, $\Phi_{\text{v-enc}}(\cdot)$, $\Phi_{\text{t-enc}}(\cdot)$ and $\Phi_{\text{fuse}}(\cdot)$, denoting visual encoder, text encoder, and fusion module, respectively. $g(\cdot)$ denotes a group of templates that decorate each of the visual categories in the text prompt, *e.g.*, 'a photograph of a [category name]'. At training time, there are totally $\mathcal{O}_i$ object categories in the text prompt, and they fall into a pre-defined set of vocabularies $\mathcal{C}_{\text{train}}$; while at inference time, we expect the model to be highly generalizable, that should equally be capable of segmenting objects from unseen categories, *i.e.*, $\mathcal{C}_{\text{test}} \supset \mathcal{C}_{\text{train}}$.

In the following sections, we start by introducing the procedure for automatically constructing a training set in Sec. 4.1, followed by the architecture design for open-vocabulary grounding in Sec. 4.2, lastly, we detail the visual instruction tuning procedure, that trains the model with only a handful of image-segmentation pairs as visual demonstrations in Sec. 4.3.

4.1. Dataset Construction

Here, we introduce a novel idea to automatically construct {visual feature, segmentation, text prompt} triplets, that are used to train our proposed grounding module. In practise, we first prepare a vocabulary with common object categories, *e.g.*, the classes in PASCAL VOC can form a category set as $\mathcal{C}_{\text{pascal}} = \{\text{'dog'}, \text{'cat'}, \ldots\}$, $|\mathcal{C}_{\text{pascal}}| = 20$, then we randomly select a number of classes to construct text prompts for image generation (*e.g.*, 'a photograph of a dog and cat'). Repeating the above operation, we can theoretically generate an infinite number of image (intermediate visual representation, *i.e.*, $\mathcal{F}$) and text prompt pairs. To acquire the segmentation masks, we take an off-the-shelf object detector, *e.g.*, pre-trained Mask R-CNN [21], and run the inference procedure on the generated images:

$$m_i^{\text{gt}} = \Phi_{\text{detector}}(\mathcal{I}_i), \quad \text{where } \mathcal{I}_i = \Phi_{\text{diffusion}}(\epsilon, y_i), \qquad (4)$$

where $m_i^{\text{gt}} \in \{0,1\}^{H \times W \times \mathcal{O}_i}$ refers to the predicted masks for a total of $\mathcal{O}_i$ objects in the generated image $\mathcal{I}_i$, conditioning on the text prompt y_i.

To evaluate the effectiveness of our proposed induction procedure for open-vocabulary grounding, we divide the vocabulary into seen categories ($\mathcal{C}_{\text{seen}}$) and unseen categories ($\mathcal{C}_{\text{unseen}}$), the training set ($\mathcal{D}_{\text{train}}$) only has images of seen categories, and the test set ($\mathcal{D}_{\text{test}}$) has both seen and unseen categories. **Note that**, in addition to using the off-the-shelf detector to generate oracle masks and construct dataset, we also explore to utilize **real public dataset**, *e.g.* PASCAL VOC, to train our grounding module via DDIM inverse process. We show the details in Sec. **??**.

4.2. Architecture

Given one specific training triplet, we now detail three components in the proposed grounding module: visual encoder, text encoder, and fusion module.

Visual Encoder. Given the visual representation, *i.e.*, latent features from the Stable Diffusion, we concatenate features with the same spatial resolution (from U-Net encoding and decoding path) to obtain $\{f_i^1, \ldots, f_i^n\}$, where $f_i^k \in \mathbb{R}^{\frac{h}{2^k} \times \frac{w}{2^k} \times d_k}$, $k \in \{0, \ldots, n\}$ denotes layer indices of U-Net, d_k refers to the feature dimension.

Next, we input $\{f_i^1, \ldots, f_i^n\}$ to visual encoder for generating the fused visual feature $\hat{\mathcal{F}}_i = \Phi_{\text{v-enc}}(\{f_i^1, \ldots, f_i^n\})$. As shown in Fig 3, visual encoder consists of 4 types of blocks: (1) 1×1 `Conv` for reducing feature dimensionality, (2) `Upsample` for upsampling the feature to a higher spatial resolution, (3) `Concat` for concatenating features, and (4) `Mix-conv` for blending features from different spatial resolutions, that includes two 3×3 `Conv` operations with residual connection and a `conditional batchnorm` operation, similar to [17].

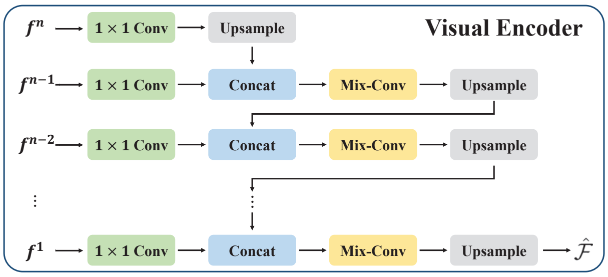


Figure 3: **The architecture of visual encoder.** The features extracted from U-Net are first grouped according to their resolution, then gradually upsampled and fused into the final visual feature.

Text Encoder. We adopt the language model from pre-trained Stable Diffusion, specifically, given the text prompt y_i, the text encoder output the corresponding embeddings for all visual objects: $\mathcal{E}_{\text{obj}_i} = \Phi_{\text{t-enc}}(g(y_i))$.

Fusion Module. The fusion module computes interaction between visual and text embeddings, then outputs segmentation masks for all visual objects. In specific, we use a standard transformer decoder with three layers, the text embeddings are treated as `Query`, that iteratively attend the visual feature for updating, and are further converted into per-segmentation embeddings with a Multi-Layer Perceptron (MLP). The object segmentation masks can be obtained by dot producting visual features with the per-segmentation embeddings. Formally, the procedure can be denoted as:

$$\mathcal{E}_{\text{seg}_i} = \Phi_{\text{TRANSFORMER-D}}(W^Q \cdot \mathcal{E}_{\text{obj}_i}, W^K \cdot \hat{\mathcal{F}}_i, W^V \cdot \hat{\mathcal{F}}_i) \quad (5)$$

$$m_i = \hat{\mathcal{F}}_i \cdot [\Phi_{\text{MLP}}(\mathcal{E}_{\text{seg}_i})]^T \quad (6)$$

where the transformer decoder generates per-segmentation embedding $\mathcal{E}_{\text{seg}_i} \in \mathbb{R}^{N \times d_e}$ for all visual objects described in the text prompt, W^Q, W^K, W^V refer to the learnable parameters for `Query`, `Key` and `Value` projection.

4.3. Training

With the constructed dataset, we can now start supervised training the proposed grounding module:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[m_i^{\text{gt}} \cdot \log(\sigma(m_i)) + (1 - m_i^{\text{gt}}) \cdot \log(\sigma(1 - m_i))]$$

where $m_i^{gt} \in \mathbb{R}^{H \times W \times \mathcal{O}_i}$ refers to the oracle groundtruth segmentation from the off-the-shelf object detector, and $m_i \in \mathbb{R}^{H \times W \times \mathcal{O}_i}$ refers to the predicted segmentation from our grounding module, $\sigma(\cdot)$ refers to Sigmoid function.

In practise, while using the off-the-shelf detector to generate segmentation masks, the model may sometimes fail to

Test Setting		**PASCAL-sim**					**COCO-sim**				
	# Objects	One		Two			One		Two		
	Categories	Seen	Unseen	Seen	Seen +Unseen	Unseen	Seen	Unseen	Seen	Seen +Unseen	Unseen
DAAM [31]	Split1	61.66	75.63	46.74	51.31	69.94	62.25	55.56	49.68	52.06	43.35
	Split2	65.75	59.25	49.08	47.98	41.50	60.08	65.55	48.80	54.66	33.22
	Split3	67.11	53.82	48.80	48.28	41.41	62.81	52.48	50.85	49.84	45.80
	Average	64.84	62.90	48.21	49.19	50.95	61.71	57.76	49.78	52.19	40.79
Ours	Split1	90.16	83.19	78.93	66.07	57.93	83.35	76.81	64.64	57.15	47.77
	Split2	90.08	86.19	78.68	67.10	47.21	82.83	74.93	63.39	57.18	42.82
	Split3	90.67	79.86	79.68	70.42	62.07	84.85	67.89	65.70	54.60	42.62
	Average	**90.30**	**83.08**	**79.10**	**67.86**	**55.74**	**83.68**	**73.21**	**64.16**	**56.31**	**44.40**

Table 1: **Quantitative result for Protocol-I evaluation on grounded generation.** Our model has been trained on the synthesized training dataset, that consists of images with one or two objects from only seen categories, and test on our synthesized test dataset that consists of images with one or two objects of both seen and unseen categories. Split1, Split2 and Split3 refer to 3 different splits of $\mathcal{C}$ that construct 3 different $(\mathcal{C}_{\text{seen}}, \mathcal{C}_{\text{unseen}})$ pairs, respectively. Our model outperforms the DAAM [31], by a large margin, see text for more detailed discussion.

detect the objects mentioned in the text prompt, and output incorrect segmentation masks. Such error comes from two sources, (i) the diffusion model may fail to generate high-quality images; (ii) off-the-shelf detector may fail to detect the objects in the synthetic image, due to the domain gap between synthetic and real images. Here, we consider two training strategies, **Normal Training**, where we fully trust the off-the-shelf detector, and use all predicted segmentation masks to train the grounding module; alternatively, we also try **Training w.o. Zero Masks**, as we empirically found that the majority of failure cases come from false negatives, that is to say, the detector failed to detect the objects and output an all-zero mask, therefore, we can train the grounding modules by ignoring the all-zero masks.

5. Experiments

In this section, we present the evaluation detail for validating the effectiveness of grounding module and its usefulness for training discriminative models, specifically, we consider two protocols: in Sec. 5.1, we train the grounding module with the constructed training set, and test the segmentation performance on synthetically generated images from Stable Diffusion, then compare with a strong off-the-shelf object detector; in Sec. 5.2, we use our augmented diffusion model to construct a synthesized semantic segmentation dataset and train a semantic segmentation model for zero-shot segmentation. Lastly, in Sec. 5.3, we conduct a series of ablation studies on the different training strategies and effects on the number of seen categories.

5.1. Protocol-I: Open-vocabulary Grounding

Here, we train the grounding module with our constructed training set, as described in Sec. 4.1. Specifically, the training set only consists of a subset of common (seen) categories, while the test set consists of both seen and unseen categories. In the following, we describe the implementation and experimental results in detail, to thoroughly assess the model on open-vocabulary grounding.

Following the dataset construction procedure as introduced in Sec. 4.1, we make **PASCAL-sim** and **COCO-sim**, with the same category split of PASCAL VOC [8] and COCO [19] as in [10, 35, 7]: (i) PASCAL-sim contains 15 seen and 5 unseen categories respectively; (ii) COCO-sim has 65 seen and 15 unseen categories respectively.

Training Set. For PASCAL-sim or COCO-sim, we generate a synthetic training set by randomly sampling images from pre-trained Stable Diffusion. This exposes our grounding module to a great variety of data ($>$ 40k) at training time, and the model is unlikely to see the same labeled examples twice during training. Unlike BigDatesetGAN [17], where only a single object is considered, we construct the text prompt with one or two objects at a time, note that, for training, only the seen categories are sampled. Although we can certainly generate images with more than two object categories, the quality of the generated images tends to be unstable, limited by the generation ability of Stable Diffusion, thus we only consider synthesized images with less than three object categories.

Testing Set. For evaluation purpose, we generate two synthetic test sets with offline sampling for PASCAL-sim and COCO-sim, respectively. In total, we have collected about 1k images for PASCAL-sim, and about 5k images for COCO-sim, we run the off-the-shelf object detector on these generated images to produce the oracle segmentation. For both test sets, the images containing two categories will be divided into three groups: both seen, both unseen, one seen and one unseen. We leave the detailed statistics of our synthetic dataset in the supplementary material. **Note that**, we have manually checked all the images and the segmentation produced from the off-the-shelf detector, and only keep the high-quality ones, thus the performance evaluation of our grounding module can be a close proxy.

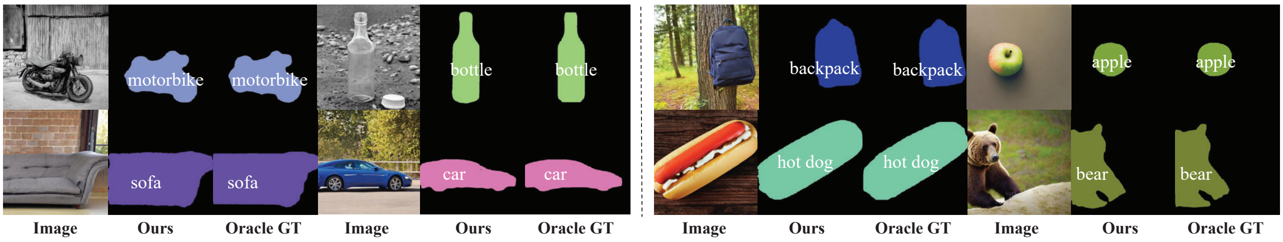


Figure 4: **Segmentation results of PASCAL-sim (left) and COCO-sim (right)** on seen (motorbike, bottle, backpack, and apple) and unseen (sofa, car, hot dog, and bear) categories. Our grounded generation model achieves comparable segmentation results to the oracle groundtruth generated by the off-the-shelf object detector.

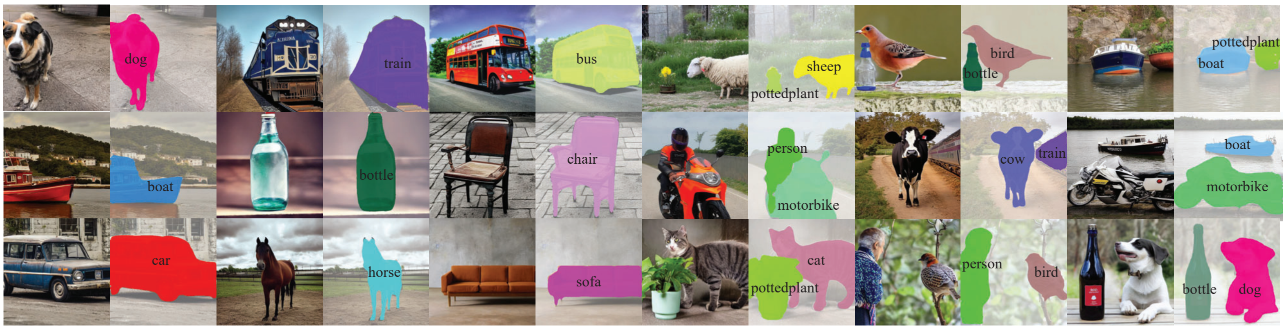


Figure 5: **Our synthesized semantic segmentation dataset** with one category (left) and two categories (right) for Protocol-II training.

Evaluation Metrics. We use the category-wise mean intersection-over-union (mIoU) as evaluation metric, defined as averages of IoU over all the categories: mIoU $= \frac{1}{C}\sum_{c=1}^{C} \text{IoU}_c$, where C is the number of all target categories, and IoU_c is the intersection-over-union for the category with index is c.

Baseline. DAAM [31] is used as a baseline for comparison, where the attention maps are directly upsampled and aggregated at each time step to explore the influence area of each word in the input text prompt.

Implementation Details. We use the pre-trained Stable Diffusion [26] and the text encoder of CLIP [24] in our implementation. We choose the Mask R-CNN [21] trained on COCO dataset as our object detector for oracle groundtruth segmentation. We fuse features from U-Net and upsample them into 512×512 spatial resolution, the text and visual embeddings are both mapped into 512 dimensions before feeding into the fusion module. We train our grounding module with two NVIDIA GeForce RTX 3090 GPUs for 5k iterations with batch size equal to 8, ADAM[14] optimiser with $\beta_1 = 0.9$, $\beta_2 = 0.999$. The initial learning rate is set to 1e-4 and the weight decay is 1e-4.

Results. As shown in Tab. 1, we provide experimental results for our grounding module, we change the composition of categories three times and compute the results for each split. Here, we can make the following observations: *first*, our model significantly outperforms the unsupervised method DAAM [31] in the mIoU on all test settings. This is because DAAM tends to result in ambiguous segmentations, as the textual embedding for individual visual entity will largely be influenced by other entities and the global sentence at the text encoding stage; *second*, our grounding module achieves superior performance on both seen and unseen categories, indicating its open-vocabulary nature, *i.e.*, the guided diffusion model can synthesize images and their corresponding segmentations for more categories beyond the vocabulary of the off-the-shelf detector.

Visualization. We demonstrate the visualization results in Fig. 4. On both seen and unseen categories, our model can successfully ground the objects in terms of segmentation mask. Impressively, as shown in Fig. 1, our grounding module can even segment the objects beyond any off-the-shelf detector can do, showing the strong open-vocabulary grounding ability of our model.

5.2. Protocol-II: Open-vocabulary Segmentation

In the previous protocol, we have validated the ability for open-vocabulary grounding on synthetically generated images. Here, we consider to adopt the images and grounding masks for tackling discriminative tasks. In particular, we first construct a synthesized image-segmentation dataset with the Stable Diffusion and our grounding module, then train a standard semantic segmentation model on such a synthetic dataset, and evaluate it on public image segmentation benchmarks.

Synthetic Dataset. To train the semantic segmentation

	PASCAL VOC					COCO				
Method	Training Set / # Categories		mIOU			Training Set / # Categories		mIOU		
	Real / 15	Synthetic / 15+5 (# Objects)	Seen	Unseen	Harmonic	Real / 65	Synthetic / 65+15 (# Objects)	Seen	Unseen	Harmonic
ZS3 [3]	✓	✗	78.0	21.2	33.3	✓	✗	-	-	-
SPNet [35]	✓	✗	77.8	25.8	38.8	✓	✗	33.0	21.9	26.3
CaGNet [10]	✓	✗	78.6	30.3	43.7	✓	✗	-	-	-
Joint [1]	✓	✗	77.7	32.5	45.9	✓	✗	**57.9**	8.6	14.9
STRICT [23]	✓	✗	82.7	35.6	49.8	✓	✗	22.2	20.4	21.3
SIGN [5]	✓	✗	83.5	41.3	55.3	✓	✗	-	-	-
ZegFormer[7]	✓	✗	**86.4**	*63.6*	*73.3*	✓	✗	*53.3*	34.5	*41.9*
Model-A (Ours)	✗	✓(one)	62.8	50.0	55.7	✗	✓(one)	28.8	32.6	30.6
Model-B (Ours)	✗	✓(two)	65.8	60.1	62.8	✗	✓(two)	37.3	37.0	37.1
Model-C (Ours)	✗	✓(mixture)	69.5	63.2	66.2	✗	✓(mixture)	38.3	38.1	38.2
Model-D (Ours)	✓	✓(mixture)	83.0	71.3	76.7	✓	✓(mixture)	50.0	**38.2**	43.2
Model-E (Ours) (complicated prompt)	✓	✓(mixture)	*83.4*	**74.4**	**78.7**	✓	✓(mixture)	53.4	*37.8*	**44.3**

Table 2: **Comparison with previous ZS3 methods on the test sets of PASCAL VOC and COCO.** These ZS3 methods are trained on real training sets. The results on PASCAL VOC are from [7]. And we retrained these ZS3 methods on COCO using their official codes, however, due to missing implementation details, we failed to reproduce ZS3 [3], CaGNet [10] and SIGN [5].

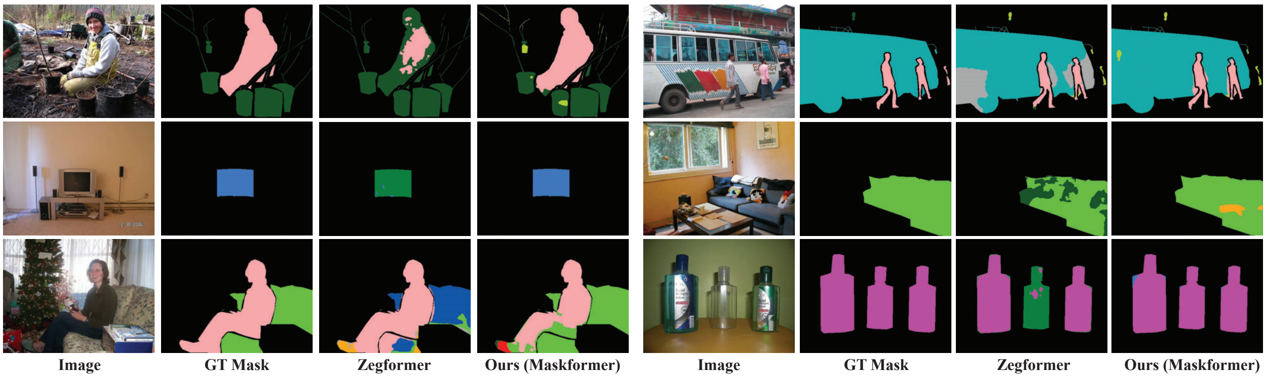


Figure 6: **Qualitative results compared with the state-of-the-art zero-shot semantic segmentation method (Zegformer) on Pascal VOC.** While training on our synthetic dataset and real dataset (only PASCAL VOC seen) together, a standard MaskFormer architecture can achieve better performance than Zegformer on unseen categories, *i.e.* pottedplant, sofa and tvmonitor. **Note that**, the synthetic data for training MaskFormer are generated from our grounding module, which has only been trained on seen categories, with no extra manual annotation involved whatsoever.

model, in addition to the provided datasets from public benchmarks, we also include two synthetically generated datasets: (i) 10k image-segmentation pairs for 20 categories in PASCAL VOC; (ii) 110k pairs for 80 categories in COCO, as shown in Fig. 5. All the image-segmentation pairs are generated by the Stable Diffusion and our proposed grounding module, that has only been trained on the corresponding seen categories (15 seen categories in PASCAL VOC and 65 seen categories in COCO). That is to say, generating these two datasets requires **no extra manual annotations** whatsoever.

Training Details. To compare with other open-vocabulary methods, our semantic segmentation model uses MaskFormer [4] with ResNet101 as its backbone. The image resolution for training is 224×224 pix, and we train the model on our synthetic dataset for 40k iterations with batch size equal to 8. We use the ADAMW as our optimizer with a learning rate of 1e-4 and the weight decay is 1e-4.

Comparison on Zero-Shot Segmentation (ZS3). While evaluating on the test sets of **real** images (1,449 images for PASCAL VOC and 5000 images for COCO), we compare with the existing zero-shot semantic segmentation approaches. As shown in Tab. 2, while only trained on synthetic dataset, our model-A,B,C have already outperformed most of ZS3 approaches on unseen categories. Specifically, the model-C trained on the mixture of one and two objects achieves the best performance. Additionally, finetuning on the real dataset with images of seen categories can

Training Type	One		Two		
	Seen	Unseen	Seen	Seen +Unseen	Unseen
Normal Training	89.88	71.18	77.66	57.24	44.22
Training w.o. Zero Masks	**90.16**	**83.19**	**78.93**	**66.07**	**57.93**

Table 3: **Ablation on training type on the constructed dataset.** Performance is measured by mIoU on **PASCAL-sim** test set.

further improve the performance, especially on seen categories (model-D). We also try to construct the synthetic dataset with more complicated prompt, *e.g.*, 'A photograph of a *bus* crossing a busy intersection in Seoul's bustling city center', and the model-E gets a slight boost on final performance. Qualitative results can be seen in Fig. 6, our model obtains accurate semantic segmentation on both seen and unseen categories.

Discussion. Overall, we can draw the following conclusions: (i) the grounding module is capable of segmenting unseen categories despite it has never seen any segmentation mask during the knowledge induction procedure, validating the strong generalisation of the grounding module; (ii) it is possible to segment more object categories by simply training on synthesized datasets, and the addition of real datasets with only seen categories can narrow the data gap thus resulting in better performance; (iii) with our proposed idea for extracting the visual-language correspondence from generative model, it introduces promising applications for applying the powerful diffusion model for discriminative tasks, *i.e.*, constructing dataset with generative model, and use it to train discriminative models, *e.g.*, expand the vocabulary of an object segmentor or detector.

5.3. Ablation study

In this section, we show the effect of different training loss and different numbers of seen categories. Due to the space limitation, we refer the reader for supplementary material, for the study on the different timestep for extracting visual representation, the number of objects in the synthetic datasets, and the effect of different datasets.

Normal Training v.s. Training without Zero Masks. As shown in Tab. 3, **Normal Training** results in unsatisfactory performance on unseen categories, we conjecture this is because the errors from detector tend to be false negative, that bias our grounding module to generate all-zero segmentation masks when encountering unseen categories; in contrast, by ignoring all-zero masks at training, **Training w.o. Zero Masks** achieves equally good performance on both seen and unseen categories.

Effect on the Number of Seen Categories. We ablate the number of seen categories to further explore the generalisation ability of our proposed grounding module. As shown in Tab. 4, the grounding module can generalise to unseen categories, even with as few as five seen categories; when introducing more seen categories, the performance on unseen ones consistently improves, but decreases on seen ones, due to the increasing complexity on seen categories.

Train Set # Seen / Unseen	One		Two		
	Seen	Unseen	Seen	Seen +Unseen	Unseen
5 / 75	**94.81**	72.42	**87.19**	49.60	39.00
20 / 60	91.91	73.33	71.59	56.27	41.91
35 / 45	87.23	73.85	66.91	55.99	43.28
50 / 30	84.55	73.20	66.41	54.39	42.71
65 / 15	83.85	**76.81**	64.64	**57.15**	**47.77**

Table 4: **Ablation on the number of seen categories on COCO-sim.** The bolded number indicates the best result. Our model can generalise to unseen categories, even as few as five seen categories.

6. Related Work

Image Generation. Image generation is one of the most challenging tasks in computer vision due to the high-dimensional nature of images. In the past few years, generative adversarial networks (GAN) [9], variational autoencoders (VAE) [16], flow-based models [15] and autoregressive models (ARM) [32] have made great progress. However, even GANs, the best of these methods, still face training instability and mode collapse issues [2]. Recently, Diffusion Probabilistic Models (DM) demonstrate state-of-the-art generation quality on highly diverse datasets [11, 22, 30, 12, 28], outperforming GANs in fidelity [6]. These models are often combined with a well-designed text encoder and trained on billions of image-caption pairs for text-to-image generation task, *i.e.*, OpenAI's DALL-E 2 [25], Google's Imagen [29] and Stability AI's Stable Diffusion [26]. However, despite being able to generate images with impressive quality using free-form text, it remains unclear what extent the visual-language correspondence has been successfully captured, this paper aims to augment an existing text-to-image diffusion model with the ability to ground objects in its generation procedure.

Visual Grounding. Visual grounding, also known as referring expression comprehension, expects to understand the natural language query and then find out the target object of the query in an image. Early visual grounding works are trained in two stages [13, 20, 33, 34, 36], by first detecting the candidate regions, and then ranking these regions. Later, one-stage approaches [18, 27, 37, 38] attract more attention due to their superior accuracy and efficiency in fusing linguistic context and visual features. Here, we consider visual grounding in the image generation procedure.

7. Conclusion

In this paper, we propose a novel idea for guiding the existing Stable Diffusion towards grounded generation, *i.e.*, segmenting the visual entities described in the text prompt while generating images. Specifically, we introduce a grounding module that explicitly aligns the visual and textual embedding space of the Stable Diffusion and train such module with an automatically constructed dataset, consisting of {image, segmentation, text prompts} triplets. Experimentally, we show that visual-language correspondence can be established by only training on a limited number of object categories, while getting the ability for open-vocabulary grounding at the image generation procedure. Additionally, we generate a synthetic semantic segmentation dataset using the augmented Stable Diffusion and train a semantic segmentation model. The model can transfer to real images and show competitive performance to existing zero-shot semantic segmentation approaches on PASCAL VOC and COCO dataset, opening up new opportunities to exploit generative model for discriminative tasks.

8. Acknowledgement

This research was supported by the National Key R&D Program of China (No. 2022ZD0161400), National Natural Science Foundation of China (62271308), STCSM (22511105700, 22DZ2229005), 111 plan (BP0719010), and State Key Laboratory of UHD Video and Audio Production and Presentation.

References

[1] Donghyeon Baek, Youngmin Oh, and Bumsub Ham. Exploiting a joint embedding space for generalized zero-shot semantic segmentation. In *Proc. ICCV*, 2021. 7

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 8

[3] Maxime Bucher, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Zero-shot semantic segmentation. *NeurIPS*, 2019. 7

[4] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *NeurIPS*, 2021. 7

[5] Jiaxin Cheng, Soumyaroop Nandi, Prem Natarajan, and Wael Abd-Almageed. Sign: Spatial-information incorporated generative network for generalized zero-shot semantic segmentation. In *Proc. ICCV*, 2021. 7

[6] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021. 8

[7] Jian Ding, Nan Xue, Gui-Song Xia, and Dengxin Dai. Decoupling zero-shot semantic segmentation. In *Proc. CVPR*, 2022. 5, 7

[8] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision(IJCV)*, 2010. 5

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 2020. 8

[10] Zhangxuan Gu, Siyuan Zhou, Li Niu, Zihan Zhao, and Liqing Zhang. Context-aware feature generation for zero-shot semantic segmentation. In *ACM MM*, 2020. 5, 7

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 8

[12] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 2022. 8

[13] Richang Hong, Daqing Liu, Xiaoyu Mo, Xiangnan He, and Hanwang Zhang. Learning to compose and reason with language tree structures for visual grounding. *IEEE transactions on pattern analysis and machine intelligence(TPAMI)*, 2019. 8

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[15] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *NeurIPS*, 2018. 8

[16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 8

[17] Daiqing Li, Huan Ling, Seung Wook Kim, Karsten Kreis, Sanja Fidler, and Antonio Torralba. Bigdatasetgan: Synthesizing imagenet with pixel-wise annotations. In *Proc. CVPR*, 2022. 4, 5

[18] Yue Liao, Si Liu, Guanbin Li, Fei Wang, Yanjie Chen, Chen Qian, and Bo Li. A real-time cross-modality correlation filtering method for referring expression comprehension. In *Proc. CVPR*, 2020. 8

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proc. ECCV*, 2014. 5

[20] Daqing Liu, Hanwang Zhang, Feng Wu, and Zheng-Jun Zha. Learning to assemble neural module tree networks for visual grounding. In *Proc. ICCV*, 2019. 8

[21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proc. ICCV*, 2021. 2, 4, 6

[22] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, 2021. 8

[23] Giuseppe Pastore, Fabio Cermelli, Yongqin Xian, Massimiliano Mancini, Zeynep Akata, and Barbara Caputo. A closer look at self-training for zero-label semantic segmentation. In *Proc. CVPRW*, 2021. 7

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 6

[25] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2, 8

[26] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image syn-

thesis with latent diffusion models. In *Proc. CVPR*, 2022. 2, 6, 8

[27] Arka Sadhu, Kan Chen, and Ram Nevatia. Zero-shot grounding of objects from natural language queries. In *Proc. ICCV*, 2019. 8

[28] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH*, 2022. 8

[29] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 2, 8

[30] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE transactions on pattern analysis and machine intelligence(TPAMI)*, 2022. 8

[31] Raphael Tang, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Jimmy Lin, and Ferhan Ture. What the daam: Interpreting stable diffusion using cross attention. *arXiv preprint arXiv:2210.04885*, 2022. 5, 6

[32] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *NeurIPS*, 2016. 8

[33] Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI)*, 2018. 8

[34] Peng Wang, Qi Wu, Jiewei Cao, Chunhua Shen, Lianli Gao, and Anton van den Hengel. Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In *Proc. CVPR*, 2019. 8

[35] Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. Semantic projection network for zero-and few-label semantic segmentation. In *Proc. CVPR*, 2019. 5, 7

[36] Sibei Yang, Guanbin Li, and Yizhou Yu. Dynamic graph attention for referring expression comprehension. In *Proc. ICCV*, 2019. 8

[37] Zhengyuan Yang, Tianlang Chen, Liwei Wang, and Jiebo Luo. Improving one-stage visual grounding by recursive subquery construction. In *Proc. ECCV*, 2020. 8

[38] Zhengyuan Yang, Boqing Gong, Liwei Wang, Wenbing Huang, Dong Yu, and Jiebo Luo. A fast and accurate one-stage approach to visual grounding. In *Proc. ICCV*, 2019. 8

[39] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022. 2