Supplementary

This supplementary material offers additional details and extended information on the token optimization used in the experiments of the paper Open-Vocabulary Attention Maps with Token Optimization for Semantic Segmentation in Diffusion Models (see Section 4), additional evaluation, and qualitative examples of the results.

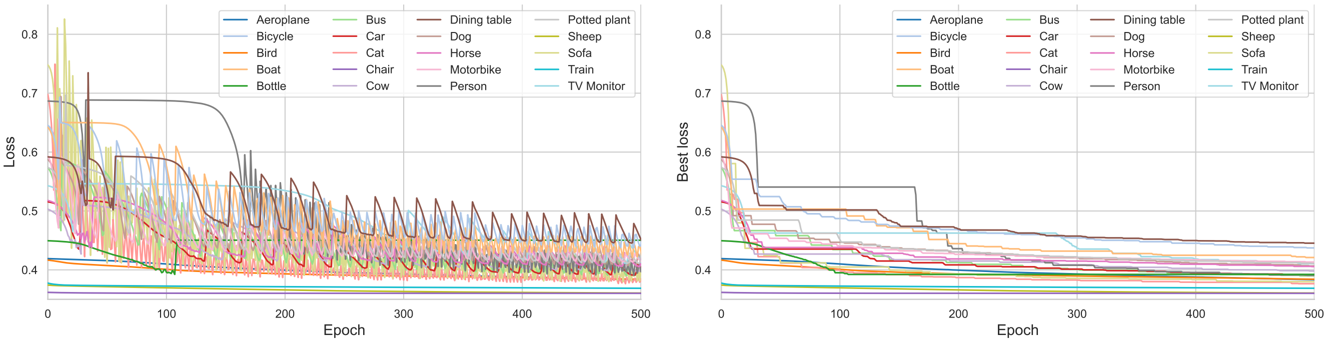
A. Implementation Details

A.1. Token Optimization via OVAM

Synthetic Images and Ground Truth Generation. To optimize a token for each of the 20 classes of the VOC Challenge [2], we generated one image per class using the text prompt *A photograph of a ⟨classname⟩*. We employed Stable Diffusion 1.5[1], the same architecture used for other OVAM experiments. We utilized 30 time steps for image generation, and default parameters of the model. The target class object in the generated image was manually annotated at a resolution of 512x512.

Initializing Token Optimization. The optimization procedure, along with other components of OVAM, is implemented in PyTorch [6]. We use gradient descent to optimize tokens. Initially, the Stable Diffusion 1.5 Text Encoder (CLIP ViT-L/14 [7]) is employed to encode the text prompt *A photograph of a ⟨classname⟩*. This encoder produces tokens with shape 1x768 and includes two special characters to mark the start and end of the text: ⟨SoT⟩ and ⟨EoT⟩. We initialize an attribution prompt, X', for optimization with tokens corresponding to ⟨SoT⟩ and the classname, forming an array of size 2x768. The ⟨SoT⟩ token is recognized for attracting background attention [9].

Performing Token Optimization. During optimization, X' is used to generate OVAM according to the methodology outlined in the paper, resulting in two attention maps of size 2x64x64. These are scaled to an image resolution of 2x512x512 using bilinear interpolation. For each channel associated with a token, binary cross-entropy is utilized to measure the discrepancy with the annotated ground truth. The loss is then backpropagated to update X'. An initial learning rate of $\alpha = 100$ is set, with a decay rate of $\gamma = 0.7$ applied every 120 steps. We run the optimization for 500 epochs, which takes less than a minute on an A40 GPU, and the best embedding is saved (Fig. S1b). Figure S1 displays the learning curves for this optimization. Despite the spiked profile of the curves by class (Fig. S1a), the procedure converges to values that generate accurate attention maps for all classes (see Fig. S3).



(a) Losses during optimization

(b) Best observed losses

Figure S1. Losses during optimization. (a) shows the losses during the training process, and (b) presents the best losses achieved.

Use of Optimized Tokens. The 20 tokens, each optimized using one annotated training image, are subsequently employed to generate attention maps for different images, thereby without the need for repeated optimization. The annex Sections A.2 and A.3 details the process of creating attention maps using these tokens and Section C provides qualitative examples of OVAM-generated maps. It also discusses the results of utilizing these tokens in conjunction with other methods.

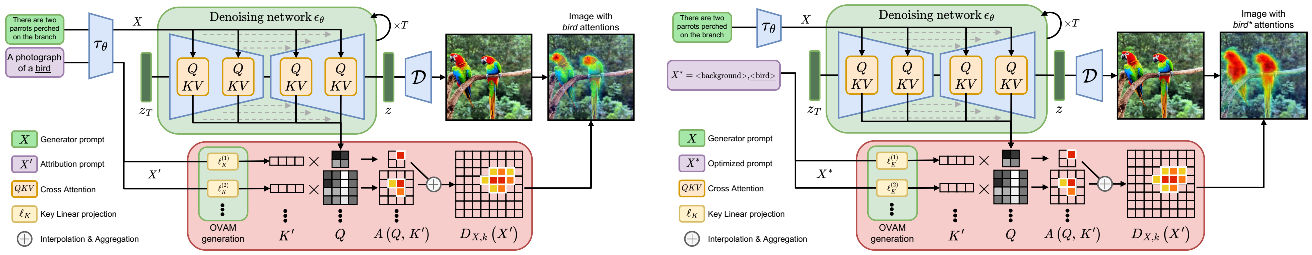
A.2. Evaluation of OVAM with Optimized Tokens

Evaluation with Natural Text. In the evaluation of OVAM attention maps with natural text (Fig. S2a), an attribution text is transformed using the Stable Diffusion 1.5 text encoder to produce a text embedding with dimensions $768 \times l_{X'}$. This embedding is then used to compute OVAM attention maps of dimensions $l_{X'} \times 64 \times 64$. Relevant maps (e.g., those corresponding to class name nouns) are extracted and resized to an image resolution of 512×512.

[1] Stable Diffusion 1.5 model card: https://huggingface.co/runwayml/stable-diffusion-v1-5 (Accessed November 2023)

Evaluation with Optimized Tokens. For the evaluation using optimized tokens (Fig. S2b), the input, shaped 2×768 (representing one token for the background and another for the class object), is utilized to compute two attention maps of dimensions $2 \times 64 \times 64$. The channel corresponding to the class object is selected and resized to form a 512×512 heatmap.

Threshold Difference. For binarizing maps generated from non-optimized tokens, a threshold of $\tau = 0.4$ is applied, followed by self-attention post-processing and dCRF. This threshold choice is based on values used in DAAM [9], a work in which OVAM's theoretical foundation is based. However, when using optimized tokens, we observe a shift in attention scale, with higher values near foreground objects (as illustrated in Fig. S3). Preliminary experiments suggest $\tau = 0.8$ as a more suitable threshold for evaluating optimized tokens.


There are two parrots perched on the branch
A photograph of a bird
τ_θ
X
Denoising network ϵ_θ
$\times T$
Q KV
z_T
z
$\mathcal{D}$
Image with *bird* attentions
X Generator prompt
X' Attribution prompt
QKV Cross Attention
ℓ_K Key Linear projection
Interpolation & Aggregation
X'
$\ell_K^{(1)}$
$\ell_K^{(2)}$
OVAM generation
K'
Q
$A(Q, K')$
$D_{X,k}(X')$
There are two parrots perched on the branch
$X^* = $ <background>,<bird>
Image with *bird** attentions
X^* Optimized prompt
X^*

(a) Evaluation of non-optimized attribution prompt

(b) Evaluation of optimized prompt

Figure S2. Comparison between the evaluation of OVAM attention maps based on (a) a natural text description, where the attentions for the word *bird* are extracted, and (b) the evaluation using an optimized token for the class *bird*.

A.3. Evaluation of other Stable Diffusion-based works

In this subsection we include further details of the evaluation of other Stable Diffusion-based works used in the experiments. Specifically, we employ DAAM [9], Attn2Mask [11], DatasetDM [10], and Grounded Diffusion [5].

Grounded Diffusion Implementation Details. Grounded Diffusion [5] extends Stable Diffusion for generating segmentation masks based on textual descriptions, by incorporating an additional trainable grounding module. This module, requiring annotated data for training, processes attentions generated during image synthesis alongside a word or token embedding. For our experiments, we utilized the official implementation available at https://github.com/Lipurple/Grounded-Diffusion, employing the weights trained with VOC classes and default setup provided by the authors. To evaluate with an optimized token, we adapted their evaluation script, allowing for direct token input instead of using a text word that is later converted into a token.

DatasetDM Implementation Details. DatasetDM [10] extends Stable Diffusion for various perception tasks, such as semantic segmentation, pose detection, and depth estimation. It includes a decoder that processes diffusion attentions and convolutional features. This decoder is trained using supervised examples. In our experiments, we employed DatasetDM's configuration for semantic segmentation along with the weights provided by the authors, trained for segmenting VOC classes. Official implementation used is available at https://github.com/showlab/DatasetDM. To evaluate optimized tokens, we modified their evaluation script to allow direct token input, instead of using a text word that is later converted.

DAAM Implementation Details. DAAM [9] is based on the direct extraction of cross-attentions during the synthesis process in Stable Diffusion. These attentions, extracted from all generation timesteps, blocks, and heads, are then aggregated and thresholded. We utilized the implementation available at https://github.com/castorini/daam, applying a threshold of $\tau = 0.4$, as recommended by the authors. For our experiments, we employed Stable Diffusion 1.5 with a 30-step generation process, aligning with the OVAM configuration. To evaluate DAAM in scenarios where the target class is not explicitly mentioned in the text prompt or using optimized tokens, we adapted DAAM to use OVAM attentions (similar adaptation illustrated in S2), which provides the same result when the word is mentioned but allowing the evaluation in all cases. For optimized tokens we use a threshold $\tau = 0.8$.

Attn2Mask. The concurrent work Attn2Mask [11] does not have any public implementation available at the time of writing this paper. Due to its similarity to OVAM without token optimization, we implemented Attn2Mask as described by the authors. For the implementation, we use Stable Diffusion 1.5 for image generation with 100 time steps. We extract cross-attentions at $t = 50$ and aggregate them. The aggregated attentions are binarized with a threshold of $\tau = 0.5$ and a dCRF [4] post-processing is applied using the SimpleCRF [3] implementation with default parameters. To evaluate optimized tokens or classes in images where the class name is not mentioned, we modify the use of cross-attention with open-vocabulary attention maps (similar adaptation illustrated in S2). For optimized tokens, we use a threshold of $\tau = 0.8$.

B. Additional Experiments

B.1. Synthetic Data Training

Extending the evaluation of the experiment in which various semantic segmentation architectures were trained using a synthetic dataset generated by OVAM (Section 4.3), this additional experiment compares the performance of optimized tokens for generating synthetic data for semantic segmentation on the VOC Challenge [2]. To generate each dataset, 1,000 synthetic images were produced using COCO captions as prompts through various Stable Diffusion extensions: DAAM [9], DatasetDM [10], Grounded Diffusion [5], and OVAM, to extract pseudo-masks. Subsequently, a Uppernet architecture with a ResNet-50 backbone was trained on these datasets, evaluated against the official VOC challenge protocol. This study further investigates the utility of optimized tokens: for each dataset, we extracted pseudo-masks using class names as descriptors for the VOC's 20 classes, comparing the outcomes with and without the use of optimized tokens. The incorporation of optimized tokens significantly enhanced mask quality (as evidenced in Figures S4 - S7), which, in turn, improved the performance of the trained segmentor across all classes when compared to the non-optimized approach (refer to Table S1). These findings affirm the value of optimized tokens in boosting the precision of synthetically generated data by these methods, enabling effective method adaptation without additional computational costs.

Method	Token Optim.	Selected classes (VOC validation set IoU %)											mIoU
		aeroplane	bicycle	bird	boat	bus	car	cat	dog	horse	person	train	
DAAM [9]	✗	27.2	16.6	41.1	21.9	56.0	37.3	32.7	30.6	25.9	23.1	38.4	27.5
	✓	**37.0**	**28.7**	**48.8**	**28.4**	**53.5**	**54.8**	**33.4**	**31.6**	**40.9**	**24.3**	**50.3**	**32.5**
DatasetDM [10]	✗	60.0	23.0	44.9	41.4	45.0	52.8	33.9	27.5	**48.9**	14.3	41.5	34.0
	✓	**60.7**	**31.8**	**51.6**	**41.7**	**58.7**	**56.8**	**37.3**	**32.7**	**48.9**	**20.1**	**56.3**	**35.5**
Grounded Diffusion [5]	✗	63.4	10.2	34.3	12.6	17.2	20.6	38.2	38.1	46.6	10.8	12.9	23.5
	✓	**67.4**	**27.1**	**43.6**	**45.9**	**63.6**	**48.8**	**42.0**	**38.6**	**48.9**	**11.3**	**44.2**	**34.9**
OVAM	✗	49.9	31.4	28.0	25.9	51.2	54.0	15.2	23.5	42.6	10.9	38.2	30.0
	✓	**57.5**	**32.2**	**44.6**	**41.1**	**58.1**	**55.2**	**42.4**	**28.0**	**44.4**	**22.4**	**51.9**	**36.1**

Table S1. Evaluation of VOC challenge performance for a model trained on synthetic data, comparing the impact of token optimization.

B.2. Presence of token in prompts

To explore the impact of explicitly mentioning the word used for extracting attentions (attribution prompt) within the image synthesis prompt (generator prompt), Table S2 expands on the overview provided in Table 1 (Section 4.1). This table breaks down the COCO-cap results by class and distinguishes between cases where the class name—used for mask generation—is included in the generator prompt or not. This detailed evaluation reveals no discernible trend to suggest that the explicit inclusion of the token in the prompt markedly influences the mIoU of the generated masks.

Method	Token included	Selected classes (COCO-cap IoU %)											mIoU
		aeroplane	bicycle	bird	boat	bus	car	cat	dog	horse	person	train	
DAAM [9]	✗	37.3	33.0	47.9	26.5	77.6	54.0	86.2	80.8	51.5	23.9	55.9	48.1
	✓	24.7	34.3	56.8	33.3	84.3	38.8	80.5	76.5	49.1	21.6	40.4	48.7
	all	30.6	33.8	53.0	31.9	82.6	42.8	83.0	77.9	49.8	22.7	44.6	48.4
DatasetDM [10]	✗	75.8	19.4	85.1	78.1	80.2	37.2	83.2	74.9	82.0	56.7	51.6	60.2
	✓	72.3	29.6	89.0	69.7	95.9	59.3	72.1	68.8	83.9	48.2	85.2	58.9
	all	74.1	25.7	87.4	71.3	91.4	51.9	76.2	71.7	83.5	52.2	72.9	59.3
Grounded Diffusion [5]	✗	84.3	47.9	80.0	61.5	94.9	0.0	89.6	83.1	86.6	53.7	67.9	52.0
	✓	85.0	58.4	83.4	17.7	80.5	27.4	85.9	85.3	83.7	49.2	44.5	47.9
	all	84.6	54.9	81.9	30.8	81.4	25.1	87.3	84.4	84.1	51.8	47.2	50.2
OVAM	✗	77.8	67.5	52.4	46.0	83.8	45.5	70.5	64.1	66.1	25.0	58.2	58.4
	✓	53.6	62.7	56.2	53.4	85.2	48.5	65.8	66.7	74.4	15.2	50.9	58.3
	all	65.1	64.3	54.6	51.9	84.9	47.5	67.9	65.8	71.5	19.7	53.0	58.2

Table S2. Table comparing mIoU whether word used for pseudo-mask generation is included in the generator prompt.

C. Qualitative Examples

C.1. Qualitative comparison of OVAM Attention Maps

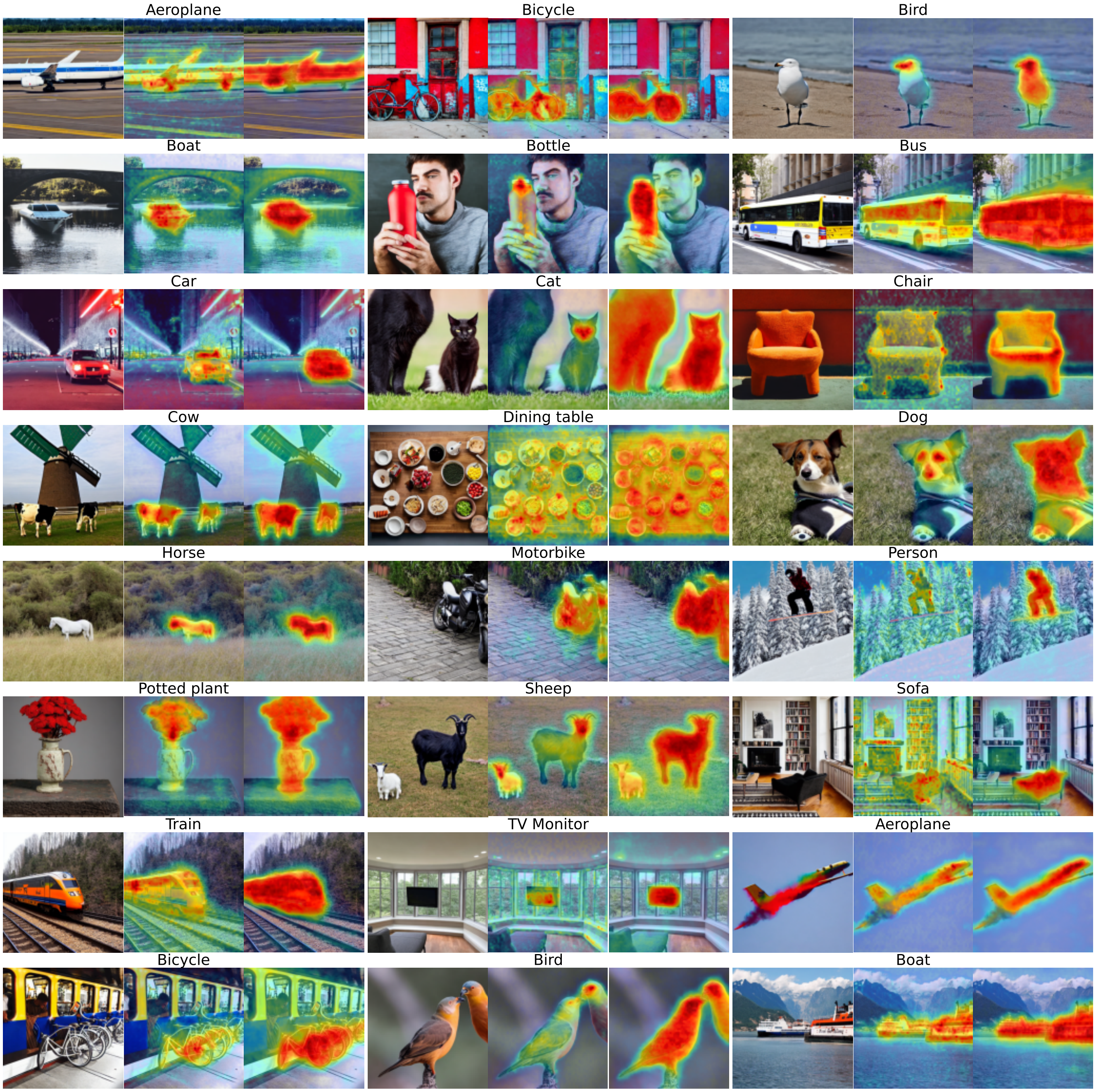


Figure S3. Qualitative Examples of synthetic images generated with Stable Diffusion 1.5 [8] and OVAM Attention Maps before binarization. For each class name, we show the obtained synthetic image (left), the attention map generated using the class name (center) and class-specific optimized tokens (right) for each of the 20 classes from the VOC challenge [2]. Images have been generated using text prompts extracted from COCO captions [1].

C.2. Use of OVAM-optimized Tokens with Other Methods

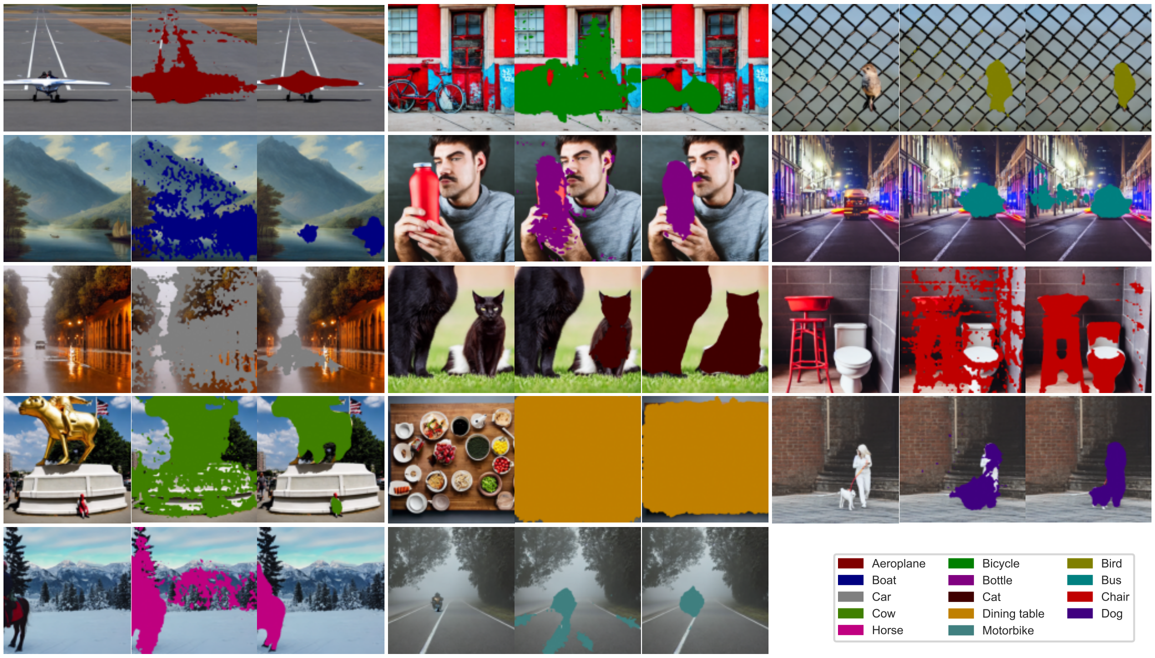


Figure S4. Qualitative Examples of **DAAM-Generated Pseudo-Masks**: Each set in the figure presents a synthetic image generated with Stable Diffusion [8] using a COCO caption [1] (left), accompanied by a mask generated through DAAM [9] using VOC class names [2] (center), and a mask generated using an OVAM-optimized token specific to the class (right).

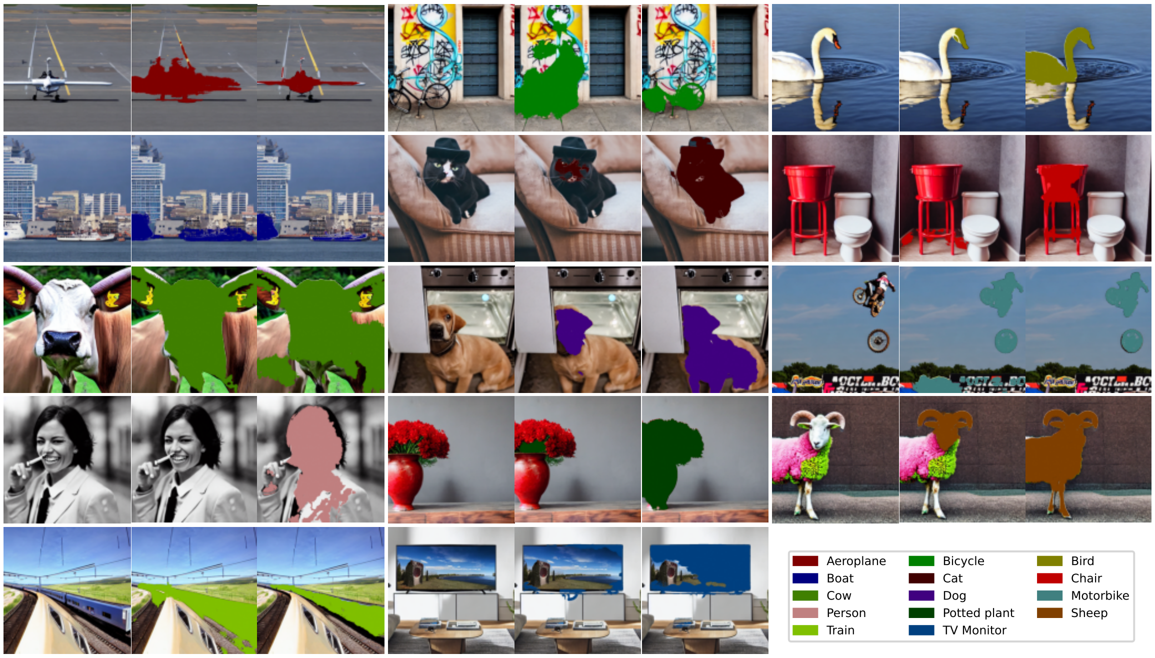


Figure S5. Qualitative Examples of **Attn2Mask-Generated Pseudo-Masks**: Each set in the figure presents a synthetic image generated with Stable Diffusion [8] using a COCO caption [1] (left), accompanied by a mask generated through Attn2Mask [11] using VOC class names [2] (center), and a mask generated using an OVAM-optimized token specific to the class (right).

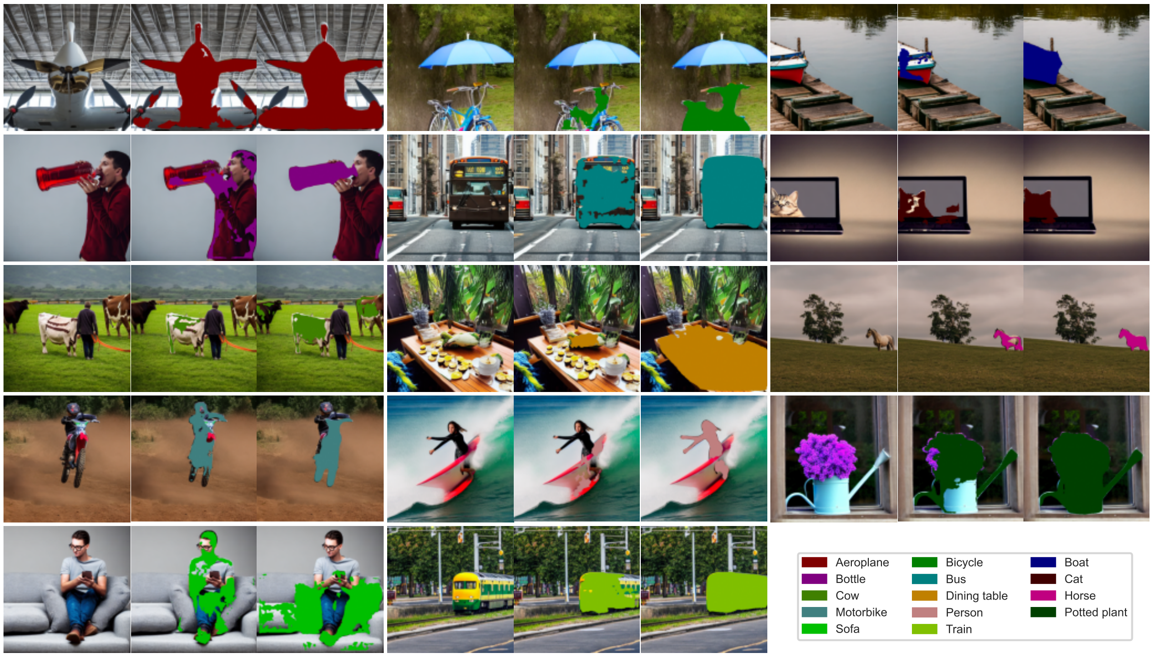


Figure S6. Qualitative Examples of **Grounded Diffusion-Generated Pseudo-Masks**: Each set in the figure presents a synthetic image generated with Stable Diffusion [8] using a COCO caption [1] (left), accompanied by a mask generated through Grounded Diffusion [5] using VOC class names [2] (center), and a mask generated using an OVAM-optimized token specific to the class (right).

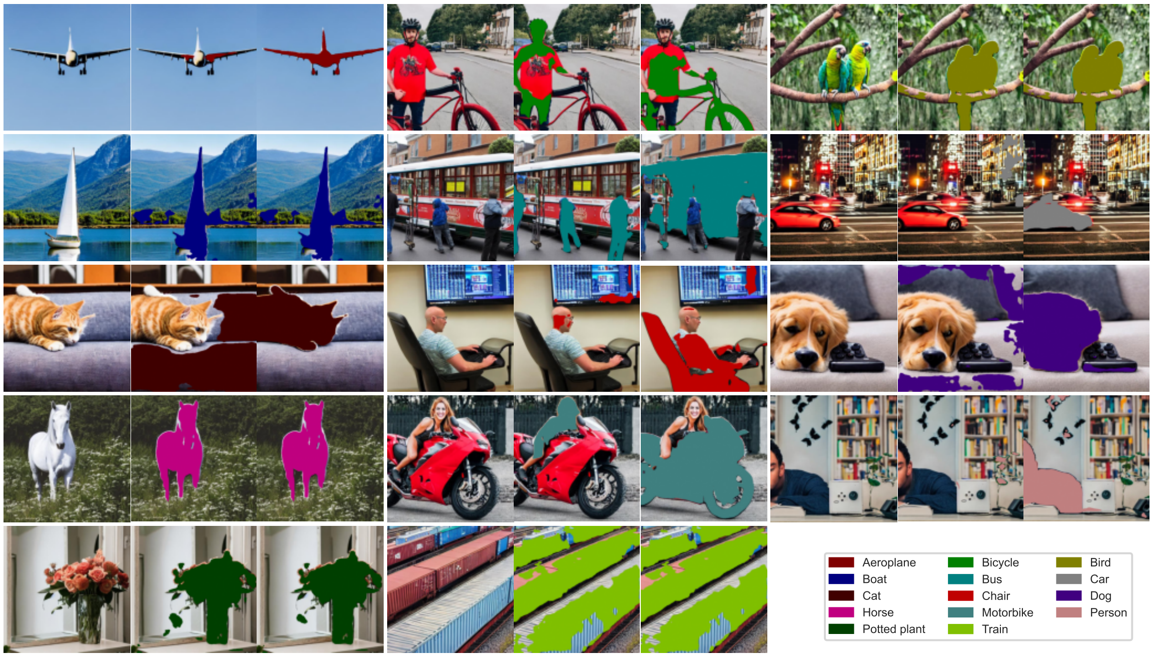


Figure S7. Qualitative Examples of **DatasetDM-Generated Pseudo-Masks**: Each set in the figure presents a synthetic image generated with Stable Diffusion [8] using a COCO caption [1] (left), accompanied by a mask generated through DatasetDM [10] using VOC class names [2] (center), and a mask generated using an OVAM-optimized token specific to the class (right). Notably, masks with non-optimized tokens sometimes segment a foreground object that does not match the intended descriptor (e.g., *cat*, *bus*, *motorbike*). The use of optimized tokens helps in aligning DatasetDM masks more accurately with the specified objects

References

[1] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 4, 5, 6

[2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 1, 3, 4, 5, 6

[3] Healthcare Intelligence Laboratory. SimpleCRF. https://github.com/HiLab-git/SimpleCRF, 2017. 2

[4] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Conference on Neural Information Processing Systems (NIPS)*, volume 24, pages 109–117, 2011. 2

[5] Ziyi Li, Qinye Zhou, Xiaoyun Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. Open-vocabulary object segmentation with diffusion models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7667–7676, 2023. 2, 3, 6

[6] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Conference on Neural Information Processing Systems (NIPS)*, pages 8024–8035, 2019. 1

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. 1

[8] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 4, 5, 6

[9] Raphael Tang, Linqing Liu, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Pontus Stenetorp, Jimmy Lin, and Ferhan Ture. What the DAAM: Interpreting stable diffusion using cross attention. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 1, pages 5644–5659, 2023. 1, 2, 3, 5

[10] Weijia Wu, Yuzhong Zhao, Hao Chen, Yuchao Gu, Rui Zhao, Yefei He, Hong Zhou, Mike Zheng Shou, and Chunhua Shen. Datasetdm: Synthesizing data with perception annotations using diffusion models. *Conference on Neural Information Processing Systems (NIPS)*, 2023. 2, 3, 6

[11] Ryota Yoshihashi, Yuya Otsuka, Kenji Doi, and Tomohiro Tanaka. Attention as annotation: Generating images and pseudo-masks for weakly supervised semantic segmentation with diffusion. *arXiv preprint arXiv:2309.01369*, 2023. 2, 5