WHAT THE DAAM: INTERPRETING STABLE DIFFUSION USING CROSS ATTENTION

Raphael Tang,[1] *Akshat Pandey,*[1] *Zhiying Jiang,*[2] *Gefei Yang,*[1] *Karun Kumar,*[1] *Jimmy Lin,*[2] *Ferhan Ture*[1]

[1]Comcast Applied AI [2]University of Waterloo

[1]firstname_lastname@comcast.com [2]{zhiying.jiang,jimmylin}@uwaterloo.ca

ABSTRACT

Large-scale diffusion neural networks represent a substantial milestone in text-to-image generation, with some performing similar to real photographs in human evaluation. However, they remain poorly understood, lacking explainability and interpretability analyses, largely due to their proprietary, closed-source nature. In this paper, to shine some much-needed light on text-to-image diffusion models, we perform a text–image attribution analysis on Stable Diffusion, a recently open-sourced large diffusion model. To produce pixel-level attribution maps, we propose DAAM, a novel interpretability method based on upscaling and aggregating cross-attention activations in the latent denoising subnetwork. We support its correctness by evaluating its unsupervised semantic segmentation quality on its own generated imagery, compared to supervised segmentation models. We show that DAAM performs strongly on COCO caption-generated images, achieving an mIoU of 61.0, and it outperforms supervised models on open-vocabulary segmentation, for an mIoU of 51.5. We further find that certain parts of speech, like punctuation and conjunctions, influence the generated imagery most, which agrees with the prior literature, while determiners and numerals the least, suggesting poor numeracy. To our knowledge, we are the first to propose and study word–pixel attribution for interpreting large-scale diffusion models. Our code and data are at `https://github.com/castorini/daam`.

***Index Terms*—** stable diffusion, explainable AI, diffusion model attribution.

1. INTRODUCTION

Diffusion neural networks trained on billions of image–caption pairs represent the state of the art in text-to-image generation [1], some achieving realism comparable to actual photographs in human evaluation. Google's Imagen, for example, produces images rated as more photorealistic than are real pictures up to 39.2–43.6% of the time [2], outperforming OpenAI's DALL-E 2 [3] in zero-shot text-to-image generation. However, despite their quality and popularity, the dynamics of their image synthesis process remain undercharacterized. Citing ethical concerns, these organizations have restricted the general public from using the models and their weights, preventing effective white-box (or even black-box) analysis. To overcome this barrier, Stability AI recently open-sourced Stable Diffusion [4], a 1.1 billion-parameter latent diffusion model pretrained and fine-tuned on the LAION 5-billion image dataset [5].

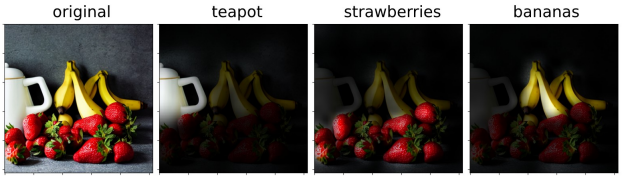

Fig. 1. The original synthesized image and three DAAM maps for teapot, strawberries, and bananas, from the prompt, "strawberries and bananas beside a teapot."

Given this opportune development, we probe Stable Diffusion to provide some much-desired insight into large diffusion models. We specialize in text-to-image attribution, our central research question being, *"Which parts of a generated image does an input word influence most?"* That is, we seek to produce a two-dimensional attribution map across the synthesized image for each word in the input prompt. As a byproduct, answering this also yields an unsupervised semantic segmentation technique for synthetic images, through extracting and attributing all nouns in the input. For example, given the phrase, "**Strawberries** next to **teapots**," we can construct pixel-level maps for "strawberries" and "teapots."

To derive these maps, we dissect the denoising autoencoder in diffusion models, where most of the synthesis occurs. In this subnetwork, attention mechanisms cross-contextualize text embeddings with coordinate-aware latent representations [4] of the image, outputting scores for each token–image patch pair. Attention scores lend themselves readily to interpretation [6] since they are already normalized in $[0, 1]$, an inherent benefit for us. Thus, for pixel-wise attribution, we propose to aggregate these scores over the spatiotemporal dimensions and upscale them across the final image—see Figure 1 for an example output. We call our method diffusion attentive attribution maps, or DAAM for short.

For evaluation, we generate images alongside DAAM maps using image captions, manually annotate object segments, then compare DAAM maps with the annotated seg-

ments. We show that, without explicit supervision, DAAM attains strong baseline quality on limited-vocabulary semantic segmentation and outperforms supervised, closed-vocabulary models on open-domain segmentation. We further apply DAAM to characterize pixel attribution for various parts of speech, finding that punctuation and conjunctions influence more of the image, while determiners and numerals the opposite, which suggests poor numeracy.

In summary, our contributions are as follows: **(1)** we propose a novel attribution method for interpreting diffusion models, targeted at measuring which parts of the generated image the words influence most; **(2)** we are the first to derive and evaluate an unsupervised open-vocabulary semantic segmentation approach for generated images; and **(3)** we provide new insight into how part of speech relates to the images.

2. OUR APPROACH

2.1. Preliminaries

Latent diffusion models [4] are a class of denoising generative models that are trained to synthesize high-fidelity images from random noise through a gradual denoising process, optionally conditioned on text. They generally comprise three components: a deep language model like CLIP [7] for producing word embeddings; a variational autoencoder (VAE) which encodes and decodes latent vectors for images; and a time-conditional U-Net [8] for gradually denoising latent vectors. To generate an image, we initialize the latent vectors to random noise, feed in a conditioning text prompt, then iteratively denoise the latent vectors with the U-Net and decode the final vector into an image with the VAE.

Formally, given an image, the VAE encodes it as a latent vector $\boldsymbol{\ell}_{t_0} \in \mathbb{R}^d$. Define a forward "noise injecting" Markov chain $p(\boldsymbol{\ell}_{t_i}|\boldsymbol{\ell}_{t_{i-1}}) := \mathcal{N}(\boldsymbol{\ell}_{t_i}; \sqrt{1-\alpha_{t_i}}\boldsymbol{\ell}_{t_0}, \alpha_{t_i}\boldsymbol{I})$ where $\{\alpha_{t_i}\}_{i=1}^T$ is defined following a schedule so that $p(\boldsymbol{\ell}_{t_T})$ is approximately zero-mean isotropic. The corresponding denoising reverse chain is then parameterized as

$$p(\boldsymbol{\ell}_{t_{i-1}}|\boldsymbol{\ell}_{t_i}) := \mathcal{N}(\boldsymbol{\ell}_{t_{i-1}}; \tfrac{1}{\sqrt{1-\alpha_{t_i}}}(\boldsymbol{\ell}_{t_i} + \alpha_{t_i}\epsilon_\theta(\boldsymbol{\ell}_{t_i}, t_i)), \alpha_{t_i}\boldsymbol{I}), \quad (1)$$

for some denoising neural network $\epsilon_\theta(\boldsymbol{\ell}, t)$ with parameters θ. Intuitively, the forward process iteratively adds noise to some signal at a fixed rate, while the reverse process, equipped with a neural network, removes noise until recovering the signal. To train the network, given caption–image pairs, we optimize

$$\min_\theta \sum_{i=1}^{T} \zeta_i \mathbb{E}_{p(\boldsymbol{\ell}_{t_i}|\boldsymbol{\ell}_{t_0})} \|\epsilon_\theta(\boldsymbol{\ell}_{t_i}, t_i) - \nabla_{\boldsymbol{\ell}_{t_i}} \log p(\boldsymbol{\ell}_{t_i}|\boldsymbol{\ell}_{t_0})\|_2^2, \quad (2)$$

where $\{\zeta_i\}_{i=1}^T$ are constants computed as $\zeta_i := 1 - \prod_{j=1}^{i}(1-\alpha_j)$. The objective is a reweighted form of the evidence lower bound (ELBO) for score matching [9]. To generate a latent vector, we initialize $\hat{\boldsymbol{\ell}}_{t_T}$ as Gaussian noise and iterate

$$\hat{\boldsymbol{\ell}}_{t_{i-1}} = \frac{1}{\sqrt{1-\alpha_{t_i}}}(\hat{\boldsymbol{\ell}}_{t_i} + \alpha_{t_i}\epsilon_\theta(\hat{\boldsymbol{\ell}}_{t_i}, t_i)) + \sqrt{\alpha_{t_i}} z_{t_i}. \quad (3)$$

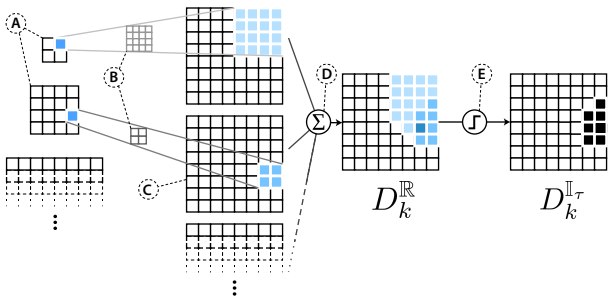

Fig. 2. Illustration of computing DAAM for some word: the multiscale attention arrays from Eqn. (5) (see **A**); the deconvolution filters (see **B**) resulting in expanded maps (**C**) from Eqn. (6); summing the heat maps across the layers (**D**), as in Eqn. (7); and the hard threshold operator (**E**) from Eqn. (8).

In practice, we apply various optimizations to improve the convergence of the above step, like modeling it as an ODE [9], but this definition suffices for us. We can additionally condition the latent vectors on text and pass word embeddings $\boldsymbol{X} := [\boldsymbol{x}_1; \cdots; \boldsymbol{x}_{l_W}]$ to $\epsilon_\theta(\boldsymbol{\ell}, t; \boldsymbol{X})$. Finally, we use the VAE decoder to decode the denoised latent $\hat{\boldsymbol{\ell}}_{t_0}$ to an image.

2.2. Diffusion Attentive Attribution Maps

Given a large-scale latent diffusion model for text-to-image synthesis, which parts of a generated image does each word influence most? Attribution approaches in computer vision are primarily perturbation- and gradient-based [10, 11], where saliency maps are either constructed from the first derivative of the output with respect to the input, or from input perturbation to see how the output changes. Unfortunately, gradient methods prove intractable from needing a backpropagation pass *for every pixel for all T time steps*, and perturbation created very different imagery in our pilot experiments. Instead, we extend analyses from natural language processing, where attention was found to indicate attribution [6].

We turn our attention to the denoising network $\epsilon_\theta(\boldsymbol{\ell}, t; \boldsymbol{X})$ responsible for the synthesis. While it can take any form, U-Nets remain the popular choice [8] for their strong image segmentation ability. They consist of a series of downsampling convolutional blocks, each of which preserves some local context, followed by upsampling deconvolutional blocks, which restore the original input size to the output. Specifically, given a 2D latent $\boldsymbol{\ell}_t \in \mathbb{R}^{h \times w}$, the downsampling blocks output a series of vectors $\{\boldsymbol{h}_{i,t}^{\downarrow}\}_{i=1}^K$, where $\boldsymbol{h}_{i,t}^{\downarrow} \in \mathbb{R}^{\lceil \frac{h}{c^i} \rceil \times \lceil \frac{w}{c^i} \rceil}$ for some $c > 1$. The upsampling blocks then iteratively upscale $\boldsymbol{h}_{K,t}^{\downarrow}$ to $\{\boldsymbol{h}_{i,t}^{\uparrow}\}_{i=K-1}^0 \in \mathbb{R}^{\lceil \frac{h}{c^i} \rceil \times \lceil \frac{w}{c^i} \rceil}$. To condition these representations on word embeddings, researchers [4] use cross-attention layers

$$\boldsymbol{h}_{i,t}^{\downarrow} := F_t^{(i)}(\hat{\boldsymbol{h}}_{i,t}^{\downarrow}, \boldsymbol{X}) \cdot (W_v^{(i)}\boldsymbol{X}), \quad (4)$$

Fig. 3. Synthetic images and their annotated segments.

$$F_t^{(i)}(\hat{\boldsymbol{h}}_{i,t}^{\downarrow}, \boldsymbol{X}) := \text{softmax}\left((W_q^{(i)}\hat{\boldsymbol{h}}_{i,t}^{\downarrow})(W_k^{(i)}\boldsymbol{X})^T/\sqrt{d}\right), \quad (5)$$

where $F_t^{(i)\downarrow} \in \mathbb{R}^{\lceil \frac{h}{c^i} \rceil \times \lceil \frac{w}{c^i} \rceil \times l_W}$ and W_k, W_q, and W_v are projection matrices. The same attention mechanism applies when upsampling $\boldsymbol{h}_i^{\uparrow}$. For brevity, we denote the respective attention score arrays as $F_t^{(i)\downarrow}$ and $F_t^{(i)\uparrow}$, and we implicitly broadcast matrix multiplications as per NumPy convention [12].

Spatiotemporal aggregation. $F_t^{(i)\downarrow}[x, y, k]$ is normalized to $[0, 1]$ and connects the k^{th} word to intermediate coordinate (x, y) for the i^{th} downsampling block. Due to the fully convolutional nature of U-Net (and the VAE), the intermediate coordinates locally map to a surrounding square *receptive field* in the final image. The scores thus relate each word to that image patch, as remarked in Hertz et al. [13], who tweak attention maps for prompt-based editing in diffusion models. However, different layers produce heat maps with varying scales, middle ones being the coarsest (e.g., $\boldsymbol{h}_{K,t}^{\downarrow}$ and $\boldsymbol{h}_{K-1,t}^{\uparrow}$), requiring spatial normalization to create a single heat map. To do this, we upscale all intermediate attention score arrays to the original image size using distribution-preserving deconvolutions

$$A_{k,t}^{(i)\downarrow} := W^{(i)} \circledast F_t^{(i)\downarrow}[:, :, k], \qquad W^{(i)}[a, b] = \left(\frac{1}{c^i}\right)^2, \quad (6)$$

for all $1 \le a \le c^i$ and $1 \le b \le c^i$, where $\circledast$ is the transposed convolution operator with stride c^i and weight $W^{(i)} \in \mathbb{R}^{c^i \times c^i}$, and $[:, :, k]$ denotes taking a slice across the height and width dimensions given k. Thus, $A_{k,t}^{(i)\downarrow}$ has size $\mathbb{R}^{h \times w}$ for all blocks i and words k, with the relative intensity preserved linearly. We can also apply a bicubic kernel for smoother maps. We derive the upsampling blocks' maps $A_{k,t}^{(i)\uparrow}$ similarly. Finally, to produce a single heat map for the k^{th} word, we sum over both the layers and the time dimension, collecting contributions across the generative iterations from Eqn. (3):

$$D_k^{\mathbb{R}}[x, y] := \sum_{j=1}^{T} \sum_{i=1}^{K} A_{k,t_j}^{(i)\downarrow}[x, y] + A_{k,t_j}^{(i)\uparrow}[x, y]. \quad (7)$$

Since $D_k^{\mathbb{R}}$ is positive and scale normalized (summing normalized values preserves linear scale), we can visualize it as a soft heat map, with higher values having greater attribution. To generate a hard, binary heat map (either a pixel is influenced or not), we can threshold $D_k^{\mathbb{R}}$ as

$$D_k^{\mathbb{I}_\tau}[x, y] := \mathbb{I}\left(D_k^{\mathbb{R}}[x, y] \ge \tau \max_{i,j} D_k^{\mathbb{R}}[i, j]\right), \quad (8)$$

where $\mathbb{I}(\cdot)$ is the indicator function and $\tau \in [0, 1]$. See Figure 2 for an end-to-end illustration of DAAM.

# Method	COCO-Gen		Unreal-Gen	
	mIoU^{80}	mIoU$^{\infty}$	mIoU^{80}	mIoU$^{\infty}$
1 Mask R-CNN (ResNet-101)	74.4	22.6	66.7	24.8
2 QueryInst (ResNet-101-FPN)	**79.4**	24.1	67.7	25.2
3 Mask2Former (Swin-S)	77.3	23.4	**72.8**	27.1
4 Random	14.1	15.0	18.3	17.2
5 Our DAAM-0.3	51.7	43.1	58.8	49.9
6 Our DAAM-0.4	53.7	**45.4**	61.0	**51.5**
7 Our DAAM-0.5	52.4	43.6	56.5	48.7

Table 1. Mean IoU of various semantic segmentation methods on our synthesized datasets. Methods before the horizontal rule are supervised on COCO, whereas those after are unsupervised, including DAAM. Best bolded.

3. EXPERIMENTS

3.1. Attribution Analysis

We first assess the veracity of DAAM as a word–pixel attribution method. For Stable Diffusion, we set classifier guidance to the default 7.5 and use 50 inference steps with PNDM [14]. We then synthesize one set of images using the validation set of the COCO image captions dataset [15], representing realistic prompts, and another set by randomly swapping nouns in the same set (holding the vocabulary fixed), representing unrealistic texts. We name the two respective sets "COCO-Gen" and "Unreal-Gen," each with 150 prompt–image pairs. To build ground-truth attribution maps, we extract all countable nouns from the prompts, then manually segment the instance of each noun in the generated image, if it exists (Figure 3).

To compute binary DAAM segmentation masks, we use Eqn. 8 with various thresholds $\tau \in \{0.3, 0.4, 0.5\}$, for each noun in the ground truth. We refer to these methods as DAAM-$\langle\tau\rangle$, e.g., DAAM-0.3. We also evaluate semantic segmentation models trained explicitly on COCO, like Mask R-CNN [16] with a ResNet-101 backbone [17], QueryInst [18] with ResNet-101-FPN [19], and Mask2Former [20] with Swin-S [21], all implemented in the MMDetection library [22]. As is standard in semantic segmentation [15], we compute the final mean intersection over union (mIoU) over the prediction–ground truth mask pairs. We denote mIoU^{80} when restricted to the 80 classes that the supervised baselines were trained on and mIoU$^{\infty}$ as the mIoU without class restriction.

Results. We present results in Table 1. As a sanity check, we compare to randomly segmenting 50% of the image for each word (row 4); unsurprisingly, this performs worst. As for τ, 0.4 works best on all splits, though it's not too sensitive ($\pm$2–5 points). The supervised models (rows 1–3) are constrained to COCO's 80 classes (the labels for the segmentation task, e.g., "cat," "cake"), while our unsupervised method (rows 5–7) is open vocabulary; thus, DAAM outperforms them by 21–27 points in mIoU$^{\infty}$ and underperforms by 6–26 points in

Fig. 4. Soft and hard DAAM maps for our case studies. On the left, a numeracy failure case for "two open laptops on a small round table." On the right, adjective visualization for "an angry, bald man doing research."

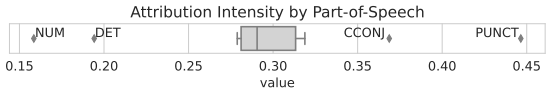

Fig. 5. Box plot of DAAM intensities for different POS tags.

mIoU^{80}. Still, DAAM forms a strong baseline of 53.7–61.0 mIoU^{80}, which is aesthetically acceptable for visualization.

On Unreal-Gen, DAAM improves in mIoU, perhaps from the scrambled nouns increasing semantic contrast (e.g., "**fox** jumps over **dog**" $\mapsto$ "**cow** jumps over **moon**") and hence "separability" in the attention maps. On the other hand, the supervised methods worsen, likely from the unrealistic imagery being semantically out of domain. Overall, we conclude that DAAM effectively constructs word–image attribution maps.

3.2. Part-of-Speech Analysis

Our attribution analysis focuses on nouns only, out of necessity to compare with semantic segmentation models. In this section, we characterize attribution patterns for other parts of speech, like adjectives and verbs, as well as punctuation. Toward this, we generate 500 images by randomly picking captions from COCO, constructing DAAM maps for every token. We group words by part-of-speech (POS), extracted by spaCy [23], then compute the average intensity of each group's DAAM maps, defined as the proportion of the image that $D_k^{\mathbb{I}_\tau}$ covers for $\tau = 0.4$, the best value in the last section.

Results. In Figure 5, we plot per-group average intensities for numerals (NUM), determiners (DET), coordinating conjunctions (CCONJ), punctuation (PUNCT), nouns, adverbs, verbs, adpositions, adjectives, and pronouns. We find two outliers at each extreme: numerals and determiners on the far left, their maps covering 16–19% of the image on average; and coordinating conjunctions and punctuation the far right, covering 37–44%. All other POS tags fall between 28–32%.

Determiners have low coverage possibly because they add little visuals, e.g., "a dog" vs. "dog." In the case of numerals, however, we conjecture that the low coverage arises from poor numeracy in Stable Diffusion, which often generates the wrong number of objects, as shown in Section 3.3. This likely arises from picking CLIP as the text encoder, which is known to suffer at numeracy [7]. On the other hand, coordinating conjunctions and punctuation have high coverage, the former from arranging objects and phrases, which could be a large portion of the image. As for punctuation, we hypothesize that punctuation aggregates information and modulate the image representation globally, as found in previous works [6].

3.3. Interpretability Case Studies

First, we present an analysis of a numeracy failure case in Figure 4, where Stable Diffusion incorrectly generates four laptops instead of two. We observe that the attribution maps reflect the laptops and the round tables (columns 2–4), but they attribute nothing for "two," suggesting a lack of attention to that word and numerals in general. This case study agrees with our findings from Section 3.2, where we show that numerals attain the least attribution across the images.

Second, we visualize adjectives–see right subfigure. The diffusion model strongly localizes "angry" and "bald" to their respective facial features, with "angry" attending to the furrowed brow and frown, and "bald" to the bare scalp. In both case studies, DAAM provides correct segments for all objects.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we present a word–pixel attribution method for text-to-image diffusion models. We apply our method to Stable Diffusion, evaluating the resulting maps using an semantic segmentation task, where we achieve strong baseline quality relative to supervised models and superiority in open-vocabulary segmentation. We find that punctuation and conjunctions attend broadly, while numerals and determiners attend little, possibly due to drawbacks in the text encoder. One promising line of future work is to extend DAAM to work in an unsupervised manner on open-vocabulary semantic segmentation on real images [24].

5. REFERENCES

[1] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, et al., "Diffusion models: A comprehensive survey of methods and applications," *arXiv:2209.00796*, 2022.

[2] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, et al., "Photorealistic text-to-image diffusion models with deep language understanding," *arXiv:2205.11487*, 2022.

[3] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen, "Hierarchical text-conditional image generation with CLIP latents," *arXiv:2204.06125*, 2022.

[4] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of CVPR*, 2022.

[5] Christoph Schuhmann, Romain Beaumont, Cade W Gordon, Ross Wightman, Theo Coombes, et al., "LAION-5B: An open large-scale dataset for training next generation image-text models," 2022.

[6] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning, "What does BERT look at? an analysis of BERT's attention," in *Proceedings of BlackboxNLP*, 2019.

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, et al., "Learning transferable visual models from natural language supervision," in *Proceedings of ICML*, 2021.

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.

[9] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole, "Score-based generative modeling through stochastic differential equations," in *Proceedings of ICLR*, 2021.

[10] David Alvarez-Melis and Tommi S Jaakkola, "On the robustness of interpretability methods," *arXiv:1806.08049*, 2018.

[11] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proceedings of ICCV*, 2017.

[12] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, et al., "Array programming with NumPy," *Nature*, 2020.

[13] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or, "Prompt-to-prompt image editing with cross attention control," *arXiv:2208.01626*, 2022.

[14] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao, "Pseudo numerical methods for diffusion models on manifolds," in *Proceedings of ICLR*, 2021.

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, et al., "Microsoft COCO: Common objects in context," in *Proceedings of ECCV*, 2014.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, "Mask R-CNN," in *Proceedings of ICCV*, 2017.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of CVPR*, 2016.

[18] Yuxin Fang, Shusheng Yang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu, "Instances as queries," in *Proceedings of ICCV*, 2021.

[19] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie, "Feature pyramid networks for object detection," in *Proceedings of CVPR*, 2017.

[20] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proceedings of CVPR*, 2022.

[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 *Proceedings of ICCV*, 2021.

[22] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, et al., "MMDetection: Open MMLab detection toolbox and benchmark," *arXiv:1906.07155*, 2019.

[23] Matthew Honnibal and Ines Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," 2017.

[24] Hang Zhao, Xavier Puig, Bolei Zhou, Sanja Fidler, and Antonio Torralba, "Open vocabulary scene parsing," in *Proceedings of ICCV*, 2017.