# Panoptic Diffusion Models: co-generation of images and segmentation maps

Yinghan Long Purdue University long273@purdue.edu

Abstract

Recently, diffusion models have demonstrated impressive capabilities in text-guided and image-conditioned image generation. However, existing diffusion models cannot simultaneously generate an image and a panoptic segmentation of objects and stuff from the prompt. Incorporating an inherent understanding of shapes and scene layouts can improve the creativity and realism of diffusion models. To address this limitation, we present Panoptic Diffusion Model (PDM), the first model designed to generate both images and panoptic segmentation maps concurrently. PDM bridges the gap between image and text by constructing segmentation layouts that provide detailed, built-in guidance throughout the generation process. This ensures the inclusion of categories mentioned in text prompts and enriches the diversity of segments within the background. We demonstrate the effectiveness of PDM across two architectures: a unified diffusion transformer and a two-stream transformer with a pretrained backbone. We propose a Multi-Scale Patching mechanism to generate high-resolution segmentation maps. Additionally, when ground-truth maps are available, PDM can function as a text-guided image-to-image generation model. Finally, we show that PDM achieves state-of-the-art results in image generation with implicit scene control.

## **1. Introduction**

Diffusion models have recently outperformed other generative models, demonstrating a strong ability to generate highquality, photorealistic images and creative videos with high fidelity [3, 7, 12, 22, 23, 35, 39, 41, 42, 44]. Their success has drawn significant attention to generative AI, marking it as the next frontier following the achievements of AI in classification tasks. However, text-guided image generation often lacks control over the spatial structure of the image [47]. Current diffusion models have difficulty understanding shapes of objects because the diffusion process is uniformly applied to every pixel, without regard to the segment it belongs to. As a result, they may generate objects with unKaushik Roy Purdue University kaushik@purdue.edu

realistic shapes and miss components mentioned in the text, leading to images that are perceived as artificial, as shown in the left column of Fig.4.

To address this issue, we propose teaching diffusion models to understand object shapes and scene structures through panoptic segmentation, which provides information about both countable objects in the foreground and background elements that complements text prompts [25]. Recent works, such as ControlNet, have demonstrated that using images with complex layouts as conditions, in addition to text prompts, can precisely control the generation process [47]. These studies show that image-guided generation can better align with users' specific imaginings expressed through both text and image prompts. Inspired by this, we anticipate that if diffusion models generate segmentation maps alongside images to provide inherent guidance, they can utilize spatial composition information to create more realistic images.

The co-generation of images and masks is nontrivial and challenging because it represents a dual problem. Unlike previous approaches that rely on either a clean image or a segmentation map as a stable condition to generate the other, our model tackles the complex task of simultaneously denoising both an image and its corresponding map [10, 47]. To address this, we designed a new paradigm to solve the dual diffusion problem. Compared to using predefined segmentation maps, co-generation preserves the diversity and flexibility of the images. By generating panoptic segmentation maps, Panoptic Diffusion Models (PDMs) provide intrinsic control over image generation, while the images in turn ensure that the map generation remains coherent. Since the generation of both segmentation maps and images is guided by text, the model learns the correlation between text, images, and maps. With its enhanced scene understanding capabilities, PDMs represent a significant step towards photorealistic image generation.

The major contributions are listed below:

1. We propose a unified diffusion model that generates both images and panoptic segmentation maps. This model inherently understands scene structures through collaborative training with multimodal data, requiring no priors and providing self-control.

2. We adapt the fast ODE solver for image denoising to facilitate simultaneous image and map generation. The iterative denoising of images and maps is interlinked, ensuring consistency between them.

3. We design both a one-stream PDM and a twostream model that incorporates a pretrained image generation stream. This second approach uses efficient fine-tuning techniques and extends pretrained model's capabilities by incorporating segmentation maps.

4. We propose a Multi-Scale Patching mechanism to directly generate high-resolution segmentation maps that scale up to four times the latent size, without requiring a super-resolution model.

### 2. Panoptic Diffusion

## 2.1. Preprocessing and Postprocessing of Segmentation Maps

As shown in Fig. 1, we process the panoptic segmentation maps through several steps before feeding them into the diffusion model. Instead of using a binary mask for each object, we load pixel-level panoptic annotations. In a segmentation map  $M_0$ , each pixel's value is set to the corresponding category ID if it belongs to a segment; otherwise, its value is zero. We then convert these pixel values into analog bits [9]. Analog bits are necessary because a standard diffusion model can only generate continuous data, while segmentation classes are discrete and categorical. Since the range of category ID is from 1 to 200, each pixel is represented by 8 binary bits. Prior to noise scheduling, these bits are scaled to the range [-1, 1], matching the range of the latent input to the diffusion model. To ensure that the noise can effectively flip the bits, its absolute value must exceed one. Therefore, we set the noise added to the maps as  $\epsilon_M \sim \mathcal{N}(0, 2 * \mathbf{I})$ .

Latent diffusion models use latent representations of images encoded by a variational autoencoder(VAE) as inputs. However, using a separate VAE for encoding and decoding high-resolution segmentation maps is inefficient. We address this issue by pooling and multi-scale patching. To achieve high-resolution maps and enable more precise control, we first pool the maps to match one, two, or four times the height and width of the image latents. We use min pooling to prioritize smaller category numbers, as the COCO dataset annotations categorize 1-91 as thing categories and 92-200 as stuff categories. Next, we set the patch size of the maps to be one, two, or four times that of the images. This approach ensures that, after patchifying, the sizes of the image and map features align. Given that images have three RGB channels while maps have only one channel for the category ID before preprocessing, using a larger patch size is effective for extracting hidden features from segmentation maps. Consequently, this method allows us to generate

higher-resolution maps without the need for an additional VAE or a larger latent size.

#### 2.2. Forward Diffusion Process

In the forward pass of the diffusion process [21], random noise  $\epsilon \sim \mathcal{N}(0, \mathbf{I})$  is added to the image latent  $x_0$  according to the noise scheduler. With a total of n steps, each step updates the noisy image  $x_t$  from the previous step  $x_{t-1}$ , using scaling factors  $\alpha$  and  $\beta$  provided by the noise scheduler. This process forms a Markov chain. Consequently, the noisy image  $x_t$  can be simplified and calculated directly from  $x_0$ .

$$x_t = \sqrt{\alpha_t} \cdot x_{t-1} + \beta_t \epsilon \tag{1}$$

$$x_t = \sqrt{\bar{\alpha} \cdot x_0} + \sigma_t \epsilon \tag{2}$$

where  $\alpha_t$  are close to 1 and  $\beta_t = 1 - \alpha_t$ . The cumulative factor  $\bar{\alpha} = \prod_{i=1}^t \alpha_i$ , and the noise is scaled by  $\sigma_t = \sqrt{1 - \bar{\alpha}}$ .

To learn to denoise panoptic segmentation maps, we create another random Gaussian noise  $\epsilon_M \sim \mathcal{N}(0, 2 * \mathbf{I})$  and add it to the ground-truth maps  $M_0$ . The same noise scheduler is used to add noises to maps.

$$M_t = \sqrt{\bar{\alpha}} \cdot M_0 + \sigma_t \epsilon_M \tag{3}$$

where  $M_t$  is the noised map at timestep t.

### 2.3. Reverse Diffusion Process

The panoptic diffusion model outputs  $\epsilon_{\theta}$ , which estimates the noise  $\epsilon$ . Using this estimated noise, we compute the predicted image  $\tilde{x}_0$ . When incorporating the map as an additional input to the diffusion model, the equation for predicting the image is given by Eq. 4. To accelerate inference, we utilize a fast DPM solver to compute  $x_{t_{i-1}}$  from  $x_{t_i}$  [29, 30]. By using discontinuous time steps  $t_i$  and  $t_{i-1}$ , this method can skip intermediate steps, reducing the total number of sampling steps required. The first-order solver is described in Equation 5, where  $h_i$  represents the difference in the log signal-to-noise ratio between different steps  $(h_i = \log(\alpha_{t_i}/\sigma_{t_i}) - \log(\alpha_{t_{i-1}}/\sigma_{t_{i-1}}))$ . Details on a thirdorder solver can be found in Appendix 6.

$$\tilde{x}_{0}(x_{t_{i}}, M_{t_{i}}, C, t_{i}) = \frac{x_{t_{i}} - \sigma_{t}\epsilon_{\theta}(x_{t_{i}}, M_{t_{i}}, C, t_{i})}{\sqrt{\bar{\alpha}}}$$
(4)  
$$x_{t_{i-1}} = \frac{\sigma_{t_{i-1}}}{\sigma_{t_{i}}} x_{t_{i}} - \alpha_{t_{i}}(e^{-h_{i}} - 1)\tilde{x}_{0}(x_{t_{i}}, M_{t_{i}}, C, t_{i})$$
(5)

The other output of a panoptic diffusion model is  $M_{\theta}$ , which is a prediction of  $M_0$ . Drawing inspiration from DPM-solver++, we use the following equation to estimate  $M_{t_{i-1}}$  from the previous step. It is important to note that the model directly estimates  $M_0$  rather than the noise added



Figure 1. Pipeline of Panoptic Diffusion Models

to the segmentation map, as predicting  $\epsilon_M$  does not provide effective guidance for the images. By training the diffusion model with panoptic segmentation maps, it incorporates intrinsic self-control into the image generation process.

$$M_{t_{i-1}} = \frac{\sigma_{t_{i-1}}}{\sigma_{t_i}} M_{t_i} - \alpha_{t_i} (e^{-h_i} - 1) M_{\theta}(x_{t_i}, M_{t_i}, C, t_i)$$

In a special case where ground truth maps are provided as conditions, the diffusion model will focus solely on predicting the images. This allows users to have customized control for generating desired images, similar to existing methods [47]. However, this approach limits the diversity of the generated images.

Since the generation of  $x_{t-1}$  and  $M_{t-1}$  relies on  $x_t$  and  $M_t$ , they form a dual problem. Improvements in the quality of the generated masks and images influence each other. Consequently, according to the scaling law, a larger diffusion model can produce more accurate masks, which in turn provides better control and further enhances image quality.

### 2.4. Dual training and generation

Let the inputs to a panoptic diffusion model at each timestep be image latent  $x_t$ , mask  $M_t$ , text condition encoded by a text encoder C, and timestep t. The conditional probability of  $x_{t-1}$  and  $M_0$  is given by

$$P(x_{t-1}, M_0 | x_t, M_t, c) = P(x_{t-1} | x_t, M_t, M_0, c) \cdot P(M_0 | x_t, M_t, c)$$
(6)

Equation 6 show that it is feasible to predict the segmantation map  $M_0$  first, then use it as a condition to predict  $x_{t-1}$ . However, when using a unified model to predict both  $x_{t-1}$ and  $M_0$ , the intermediate features already contain the segmentation information used to predict  $M_0$ . Through selfattention, the map features can inherently condition  $x_{t-1}$ . Therefore, it is reasonable to predict  $x_{t-1}$  and  $M_0$  simultaneously. By taking the logarithm of the probability, we can optimize the model by combining the losses associated with image denoising and segmentation map generation.

$$\log P(x_{t-1}, M_0 | x_t, M_t, c) = \log P(x_{t-1} | x_t, M_t, M_0, c) + \log P(M_0 | x_t, M_t, c)$$
(7)

The training algorithm is outlined in Algorithm 1. We use Mean Squared Error (MSE) loss to optimize the predicted noises for both image and segmentation map denoising. Specifically, the target for image denoising is the noise  $\epsilon$ , while the target for mask generation is the ground-truth  $M_0$ . The losses for images and maps are summed to perform gradient backpropagation. During inference, the diffusion model iteratively denoises both images and maps, as detailed in Algorithm 2.

### 2.5. Architecture of Panoptic Diffusion Models

### 2.5.1. One-stream Panoptic Diffusion Models

We first modify a U-ViT to a panoptic diffusion model [2]. We start by patchifying the map input  $M_t$  using a convolutional layer and adding positional embeddings. These map embeddings are then concatenated with the image, text, and time embeddings and processed through attention blocks. Since U-ViT treats all inputs as tokens and applies self-attention among them, the segmentation maps can be treated as tokens in the same manner. At the end of the transformer, we separate the features related to images and segmentation maps, using distinct convolutional layers to unpatchify and predict the outputs.

#### 2.5.2. Two-stream Panoptic Diffusion Models

To leverage a pretrained model as the backbone, we design a two-stream diffusion model consisting of a pretrained image stream and a segmentation map stream, as illustrated in Fig. 2. During fine-tuning, the transformer layers of the image stream are kept frozen while the map stream is adjusted. The map stream processes image features and conditions from the previous block, then concatenates them with map features. Through self-attention, the map features and image features become interrelated within the map stream. The auxiliary image feature output from the map stream is added back to the image stream via a zero-convolution layer. This setup ensures specific control over the image stream and allows gradients to be backpropagated from the loss of image generation. The zero-convolution layer has zero initial weights and no bias [47]. Unlike ControlNet, which uses only the encoder part of the map stream to generate control signals, our model employs encoder-decoder U-shaped transformers in both streams to co-generate images and segmentation maps.



Figure 2. Two-stream panoptic diffusion model. There are a pretrained image stream on the left and a fine-tuned segmentation map stream on the right.

## 3. Experiments

We train our model using the COCO2017 dataset [27], which includes both panoptic segmentation maps and image captions. The COCO2017 dataset comprises 118k training samples and 5k validation samples. Images are projected into latent space using a VAE model provided by Stable Diffusion [15, 41], while text conditions are encoded using the CLIP encoder from OpenAI (clip-vit-large-patch14) [38]. We implement both one-stream and two-stream panoptic diffusion models (PDM) based on U-ViT [2]. In contrast to commercial models with billions of parameters, our mod-

els are significantly smaller. The one-stream PDM has 45 million parameters, while the two-stream PDM has 95 million parameters. The image latent size is  $32 \times 32 \times 4$ , with a height and width of 32 and a latent channel count of 4. The segmentation map's height and width can be 32, 64, or 128, depending on the patch factor, and it has 8 channels, representing 8 analog bits after conversion. The diffusion model's output image latents are decoded by a VAE decoder to produce  $256 \times 256$  images.

Model	$FID(\downarrow)$	CLIP(↑)
GLIDE [35]	12.24	$\sim \! 28$
Imagen [42]	7.27	$\sim 27$
VQ-Diffusion [15]	13.86	-
UViT [2]	8.29	27.37
One-stream PDM	18.52	26.32
Two-stream PDM	10.99	27.53
One-stream PDM given maps	8.21	28.40
Two-stream PDM given maps	11.61	28.19

Table 1. Quantitative Evaluation Results of COCO dataset.

### 3.1. Quantitative Evaluation

In Table.1, we compare the FID and CLIP scores of our models with those of state-of-the-art methods [19] [18].

The results indicate that while our panoptic diffusion models (PDMs) are trained with a combined loss of images and segmentation maps, they achieve comparable fidelity (FID scores) and improved relevance between image and text (higher CLIP scores). This improvement is due to the enhanced connectivity between the image, text, and segmentation map. The two-stream PDM performs better due to its pretrained stream and larger number of parameters. When ground-truth maps are provided, the model performs optimally because it focuses solely on optimizing image generation.

For more ablation study, please see the supplementary materials 5.4.

# 4. Conclusion

In conclusion, we introduce the Panoptic Diffusion Model (PDM), a pioneering approach that simultaneously generates images and panoptic segmentation maps from a given prompt. Unlike previous diffusion models that either depend on pre-existing segmentation maps or generate them based on images, PDM inherently understands and constructs scene layouts during the generation process. This innovation enables PDM to produce more creative and realistic images by leveraging segmentation layouts as intrinsic guidance. This research lays the groundwork for future advancements in diffusion models, offering a robust framework for co-generation of images and segmentation maps.

# Panoptic Diffusion Models: co-generation of images and segmentation maps

Supplementary Material

## 5. Supplementary Material

## 5.1. Algorithms

Algorithm 1 Training of Panoptic Diffusion model

**Input:** Ground truth Masks  $M_0$ ; Images  $x_0$ ; Text condition C; Total number of steps T**Output:** Predicted noise  $\epsilon_{\theta}$ , Predicted mask  $M_{\theta}$  $\epsilon = \text{normal(mean=0, std=1)}$  $\epsilon_m = \text{normal(mean=0, std=2)}$  $M_0 = \text{int2bits}(M_0)$ t = randn(1,T) $x_t = \text{scheduler}(x_0, \epsilon, t)$  $M_t = \text{scheduler}(M_0, \epsilon_m, t)$  $\epsilon_{\theta}, M_{\theta} = \text{DiffusionModel}(x_t, M_t, C, t)$  $loss_x = \text{MSE}(\epsilon, \epsilon_{\theta})$  $loss_m = \text{MSE}(M_0, M_{\theta})$  $loss = loss_x + loss_m$ 

Algorithm 2 Inference of Panoptic Diffusion model using DPM solver

**Input:** Text *C*; Total number of steps *T*  **Output:**Generated image  $x_0$ , Generated mask  $M_0$   $x_t$ = normal(mean=0, std=1)  $M_t$  = normal(mean=0, std=1) Sample a set of steps *T* from n to 0 **for** *t* in *T* **do** # Run the diffusion model  $\epsilon_{\theta}, M_{\theta}$  = DiffusionModel( $x_t, M_t, C, t$ ) # Update predicted images and masks  $x_0 = \frac{x_t - \sigma_t \epsilon_{\theta}}{\sqrt{\alpha}}$   $x_t, M_t$  = dpmSolver( $x_0, M_{\theta}, X_t, M_t, t$ ) **end for** 

## 5.2. Generated images and segmentation maps

In Fig. 4, we compare the images and masks generated by PDM with images generated by U-ViT. By training with segmentation masks, PDM learns that the shape of a stop sign should be octagon, while U-ViT cannot guarantee to generate an octagon stop sign. Similarly, PDM ensures to generate correct shapes for a fire hydrant and a human. In the last row of Fig. 4, PDM generates masks for not only elephants but also for the river, while a regular diffusion model misses the required component of the text prompt.

Figure 3 displays images generated with either ground-truth



(a) Ground-truth segmentation maps





(b) Images generated based on ground-truth maps



(c) Generated maps

(d) Images co-generated with maps

Figure 3. Image-map co-generation. Prompts are: 1) a small copper vase with some flowers in it; 2) A giraffe examining the back of another giraffe; 3) A utility truck is parked in the street beside traffic cones; 4) A white yellow and blue train at an empty train station.

segmentation maps or co-generated maps. The generated maps in the bottom left show objects of the same categories and similar shapes as the ground-truth maps. The images on the right are conditioned on these segmentation maps, demonstrating the PDM's ability to generate correlated images and maps. While images generated with ground-truth maps exhibit slightly better quality, co-generation removes the need for a segmentation input and produces diverse maps and images.

### 5.3. Evaluation metric for generated maps

We propose a new metric to evaluate the quality of generated segmentation maps by measuring the difference in the number of pixels labeled as each category. While Panoptic Quality [25] uses Intersection over Union (IoU) to assess segmentation maps, this metric is not suitable to evaluate maps co-generated with images. We introduce the Mean Count Difference (MCD) metric. MCD evaluates the quality of generated maps by counting the frequency f of each category in both the ground-truth and generated maps, then



(a) "An upside down stop sign by the road."



(d) "A fire hydrant on the side of the street."



(b) PDM generated image of a stop sign.



(e) PDM generated image of a fire hydrant.



(g) "A man with a wet suit on standing on a surfboard in the water."



(j) "Several elephants walking together in a line near water."



(h) PDM generated

image of a man surf-

ing in the water.

(k) PDM generated image of several elephants near a river.



Figure 4. Left: images generated by a regular diffusion model (U-ViT) based on the text prompt. Right: images and masks generated by a Panoptic Diffusion Model based on the same text.

summing their absolute differences. This sum is divided by the total number of pixels, calculated as the product of the height and width. Given that object locations on the generated map are not fixed, comparing category frequencies rather than direct pixel values provides a more meaningful assessment. The metric ranges from [0, 2], where zero indicates identical segmentation maps and larger values indicate greater differences.

$$f = bincount(M_0); \quad f' = bincount(M_\theta)$$
$$MCD = \frac{\sum(|f - f'|)}{H * W}$$



Table 2 shows increasing the patch factor results in a higher MCD because generating higher-resolution maps with a fixed number of latents becomes more challenging. This creates a trade-off between map resolution and quality. We find that a patch factor of 2 offers the best balance, yielding the highest FID and CLIP. However, increasing the patch factor to 4 results in worse performance, suggesting that unbalanced patch sizes for maps and images are detrimental.

Model	$FID(\downarrow)$	CLIP(↑)	Patch	MCD
One-stream PDM	18.52	26.32	2	1.638
Two-stream PDM	11.29	27.08	1	1.522
	10.99	27.53	2	1.592
	30.91	25.87	4	1.638

Table 2. The effect of segmentation patch size on FID, CLIP, and MCD of generated images and masks

# 6. Fast DPM solver for segmentation maps

We modify the first order and third order DPM-solver++ to solve the image and map of the previous step given  $x_t$ ,  $M_t$  and predicted  $x_0$ ,  $M_0$  [30]. The pseudo code for the solvers are listed below. For the details of the algorithm and definition of the parameters  $\sigma$ ,  $\alpha$ ,  $\phi$ , s, please check DPMsolver++.

- def dpmFirstSolver(self, x\_0, m\_0, x\_t, m\_t):
   x\_t = (sigma\_t / sigma\_s) \* x
   +(alpha\_t \* phi\_1) \* x\_0
   #update M[t-1] based on M[t]
   m\_t = (sigma\_t / sigma\_s) \* m\_t +
   (alpha\_t \* phi\_1) \* m\_0
   return x\_t, m\_t
- def dpmThirdSolver(self, x\_t, m\_t, C, t): #First step  $x_0$ ,  $m_0 = diffusion(x_t, m_t, C, s)$  $x_s = (sigma_s 1 / sigma_s) * x$  $+(alpha_s1*phi_11)*x_0$  $m_s1 = (sigma_s1 / sigma_s) * m_t +$  $(alpha_s1*phi_11)*m_0$ #Second step x\_02, m\_02= diffusion(x\_s1,m\_s1,C,s1)  $x_{s2} = (sigma_{s2} / sigma_{s}) * x +$  $(alpha_s1*phi_12)*x_0 +$  $r2 / r1 * (alpha_s2*phi_22)*(x_02-x_0)$  $m_s2 = (sigma_s2/sigma_s) * m_t +$  $(alpha_s2*phi_12)*m_0 +$  $r2 / r1 * (alpha_s2*phi_22)*(m_02-m_0)$ #Third step  $x_03$ ,  $m_03=$  diffusion (x\_s2, m\_s2, C, s2)





(f) PDM generated mask for a fire hydrant.

(i) PDM generated

masks for person, sky,

and sea

```
x_t=(sigma_t/sigma_s)*x
+(alpha_t*phi_1)*x_0 +
(1/r2)*(alpha_t * phi_2)*(x_03-x_0)
m_t= (sigma_t/sigma_s)*m_t +
(alpha_t*phi_1)*m_0 +
(1/r2)*(alpha_t * phi_2)*(m_03-m_0)
return x_t, m_t
```

# 7. Related works

## 7.1. Diffusion Models for Image Generation

Denoising Diffusion Probabilistic Models (DDPM) use a Markov chain to gradually add scheduled noises to images in the forward process and then parameterize the transition by a neural network trained to predict the noise [21]. During inference, a diffusion model starts from random noise and gradually reverses it to reconstruct the image. A wellknown drawback of diffusion models is that they require a large number of steps to generate samples iteratively. To improve efficiency, researchers have proposed various modifications to diffusion models [34]. DDIM demonstrates that diffusion models can operate in a non-Markovian manner, resulting in shorter generative chains [45]. Additionally, distillation algorithms have been introduced to further accelerate the multi-step inference process[5, 40, 43]. We use a fast solver for our panoptic diffusion model, which is a modified version of DPM Solver++ that can solve the reverse of the diffusion process in 10-50 steps [29, 30].

The backbone neural network for a diffusion model is typically a UNet, which is composed of convolutional layers and attention blocks, or a diffusion transformer that relies solely on attention mechanisms [36, 41]. Another variant, UViT, is a type of diffusion transformer that retains skip connections, allowing later layers to access information from earlier layers, thereby enhancing alignment [2].

There are three main methods for applying conditions to a diffusion model. The first approach, used in stable diffusion, involves cross-attention between the image and the conditions [41]. The second method appends condition embeddings as tokens to the image patches [2]. The third approach uses an adaptive norm layer to integrate conditions with the hidden states [36]. In our panoptic diffusion models, we opt for the second method because the transformer can leverage self-attention to learn the relationships between images and maps, treating them as conditions for each other. During inference, we apply classifier-free guidance similar to Nichol et al. [35] and Ho and Salimans [20].

## 7.2. Image Segmentation

Object detection requires generating bounding boxes and fine-grained masks, tasks traditionally accomplished by convolutional neural networks such as Fast R-CNN [14] and Mask R-CNN [16]. In Carion et al. [8], researchers introduced the use of transformers to generate binary masks by inputting object queries. Building on this, Cheng et al. [11] proposed a collaboration between an image encoder backbone and a masked transformer to generate masks, where masked attention replaces cross attention. With advanced segmentation models like Segment Anything [26] easily segmenting images, segmentation maps hold potential as alternative or complementary training data for image generation tasks.

Recently, there has been growing interest in applying diffusion models to segmentation masks. For example, Baranchuk et al. [4] suggest that the intermediate features of diffusion models can capture semantic information useful for label-efficient segmentation. Similarly, DiffuMask [46] and Dataset Diffusion [33] generate a synthetic pair of an image and a corresponding segmentation annotation of objects using attention maps. However, directly extracting masks from attention maps lacks the ability to control the generated image in return. Unified diffusion models for image generation and segmentation has shown a potential to refine image generation, such as UniGS [37]. While the existing works focuses on semantic segmentation, our method extends to panoptic segmentation, providing both instance and semantic information. This is a crucial distinction and expands the potential applications of our model.

On the other hand, some previous studies use diffusion models for panoptic segmentation based on given images. In Chen et al. [10], a diffusion model comprising an image encoder and a mask decoder is used to extract image features and apply cross attention between these features and the masks. To address the challenge of handling discrete data with diffusion models, Chen et al. [9] proposed converting panoptic masks into analog bits during preprocessing. Our approach extends the ability of the diffusion model to co-generate pixel-level panoptic segmentation maps and images, allowing them to influence and control each other.

### 7.3. Image Guided Image Generation

Image guided image generation enables more precise control over the structure of the image and ensures faithfulness to users' illustrative inputs. The input for guidance can have various forms, such as segmentation maps and layouts [41, 47]. Stochastic Differential Editing (SDEdit) perturbs user inputs with Gaussian noises and then synthesizes images by reversing SDE [31]. They show that when the reverse SDE is not solved from the ending point but a particular timestep, the generated images can achieve a good balance between faithfulness and realism. Make-a-scene introduces scene-based conditioning for image generation by optionally providing tokens from segmentation maps [13], but this method heavily relies on explicit strategies for tackling panoptic, human, and face semantics. SpaText [1] employs CLIP [38] to convert local text prompts that describe segments into image space and concatenate to the channel dimension of noises. ControlNet can accept user inputs such as canny edges and segmentation masks for conditional control of image generation [47]. Prompt-to-prompt image editing controls the generation by cross-attention to ensure similarity between images generated from similar prompts [17]. InstructPix2Pix combines Prompt-to-prompt method with stable diffusion to generate pairs of images from pairs of captions for training, then train the model to modify image pixels following the instructions [6].

These approaches demonstrate that providing various forms of guidance can more accurately control the structure of generated images. Building on this insight, our method assumes that such guidance is crucial for enhancing image quality. Additionally, panoptic diffusion models inherently generate segmentation maps alongside images, offering built-in guidance without the need for additional user input beyond the text prompt.

## 7.4. Efficient Deep Learning

To reduce the number of trained parameters or adapt the model to a new domain, previous works have designed adaptive blocks to fine-tune convolutional neural networks or transformers [24, 28, 32]. In our two-stream panoptic diffusion model, the map stream functions similarly to an adapter. To prevent any negative impact on the pretrained weights, we employ zero-initialized convolutional blocks as proposed in Zhang et al. [47].

Unlike other works that introduce significant computational overhead to generate segmentation with images, our method maintains the efficiency by leveraging a bit encoding scheme and multi-scale patching. This allows for parallel generation of images and masks without substantial additional computational cost. We will include a comparison of the number of parameters to highlight this advantage.

## References

- [1] Omri Avrahami, Thomas Hayes, Oran Gafni, Sonal Gupta, Yaniv Taigman, Devi Parikh, Dani Lischinski, Ohad Fried, and Xi Yin. Spatext: Spatio-textual representation for controllable image generation. In 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2023. 3
- [2] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *CVPR*, 2023. 3, 4
- [3] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, Yuanzhen Li, Michael Rubinstein, Tomer Michaeli, Oliver Wang, Deqing Sun, Tali Dekel, and Inbar Mosseri. Lumiere: A space-time diffusion model for video generation, 2024. 1
- [4] Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models, 2021. 3
- [5] David Berthelot, Arnaud Autef, Jierui Lin, Dian Ang Yap, Shuangfei Zhai, Siyuan Hu, Daniel Zheng, Walter Talbott, and Eric Gu. Tract: Denoising diffusion models with transitive closure time-distillation, 2023. 3
- [6] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions, 2023. 4
- [7] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 1
- [8] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *CoRR*, abs/2005.12872, 2020. 3
- [9] Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. arXiv preprint arXiv:2208.04202, 2022. 2, 3
- [10] Ting Chen, Lala Li, Saurabh Saxena, Geoffrey Hinton, and David Fleed. A generalist framework for panoptic segmentation of images and videos. pages 909–919, 2023. 1, 3
- [11] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. 2022. 3
- [12] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *CoRR*, abs/2105.05233, 2021. 1
- [13] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scenebased text-to-image generation with human priors, 2022. 3
- [14] Ross B. Girshick. Fast R-CNN. CoRR, abs/1504.08083, 2015. 3
- [15] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. *CoRR*, abs/2111.14822, 2021. 4

- [16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. CoRR, abs/1703.06870, 2017. 3
- [17] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control, 2022. 4
- [18] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning, 2022. 4
- [19] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Advances in Neural Information Processing Systems. Curran Associates, Inc., 2017. 4
- [20] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. 3
- [21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NIPS*, 2020. 2, 3
- [22] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. arXiv preprint arXiv:2210.02303, 2022. 1
- [23] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models, 2022. 1
- [24] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. *CoRR*, abs/1902.00751, 2019. 4
- [25] Alexander Kirillov, Kaiming He, Ross B. Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. *CoRR*, abs/1801.00868, 2018. 1
- [26] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. arXiv:2304.02643, 2023.
- [27] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. 4
- [28] Yinghan Long, Indranil Chakraborty, Gopalakrishnan Srinivasan, and Kaushik Roy. Complexity-aware adaptive training and inference for edge-cloud distributed ai systems. In 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS), pages 573–583, 2021. 4
- [29] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *NeurIPS*, 2022. 2, 3
- [30] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models, 2023. 2, 3
- [31] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations, 2022. 3

- [32] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2iadapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models, 2023. 4
- [33] Quang Nguyen, Truong Vu, Anh Tran, and Khoi Nguyen. Dataset diffusion: Diffusion-based synthetic dataset generation for pixel-level semantic segmentation, 2023. 3
- [34] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *CoRR*, abs/2102.09672, 2021.
   3
- [35] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. *CoRR*, abs/2112.10741, 2021. 1, 4, 3
- [36] William Peebles and Saining Xie. Scalable diffusion models with transformers. arXiv preprint arXiv:2212.09748, 2022.
   3
- [37] Lu Qi, Lehan Yang, Weidong Guo, Yu Xu, Bo Du, Varun Jampani, and Ming-Hsuan Yang. Unigs: Unified representation for image generation and segmentation, 2023. 3
- [38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya 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, 2021. 4
- [39] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. 1
- [40] Yuxi Ren, Xin Xia, Yanzuo Lu, Jiacheng Zhang, Jie Wu, Pan Xie, Xing Wang, and Xuefeng Xiao. Hyper-sd: Trajectory segmented consistency model for efficient image synthesis, 2024. 3
- [41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *CVPR*, abs/2112.10752, 2022. 1, 4, 3
- [42] 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, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022. 1, 4
- [43] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *ICLR*, abs/2202.00512, 2022. 3
- [44] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data, 2022. 1
- [45] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ICLR*, abs/2010.02502, 2021.
   3
- [46] Weijia Wu, Yuzhong Zhao, Mike Zheng Shou, Hong Zhou, and Chunhua Shen. Diffumask: Synthesizing images with

pixel-level annotations for semantic segmentation using diffusion models, 2024. 3

[47] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 1, 3, 4