UniGS: Unified Representation for Image Generation and Segmentation

Lu Qi[1,*], Lehan Yang[2*], Weidong Guo[3†], Yu Xu[3],
Bo Du[4], Varun Jampani[5], Ming-Hsuan Yang[1,6],

[1]The University of California, Merced [2]The University of Sydney
[3]QQ Brower Lab, Tencent, [4]Wuhan University
[5]Stability AI [6]Google Research

Abstract

This paper introduces a novel unified representation of diffusion models for image generation and segmentation. Specifically, we use a colormap to represent entity-level masks, addressing the challenge of varying entity numbers while aligning the representation closely with the image RGB domain. Two novel modules, including the location-aware color palette and progressive dichotomy module, are proposed to support our mask representation. On the one hand, a location-aware palette guarantees the colors' consistency to entities' locations. On the other hand, the progressive dichotomy module can efficiently decode the synthesized colormap to high-quality entity-level masks in a depth-first binary search without knowing the cluster numbers. To tackle the issue of lacking large-scale segmentation training data, we employ an inpainting pipeline and then improve the flexibility of diffusion models across various tasks, including inpainting, image synthesis, referring segmentation, and entity segmentation. Comprehensive experiments validate the efficiency of our approach, demonstrating comparable segmentation mask quality to state-of-the-art and adaptability to multiple tasks.

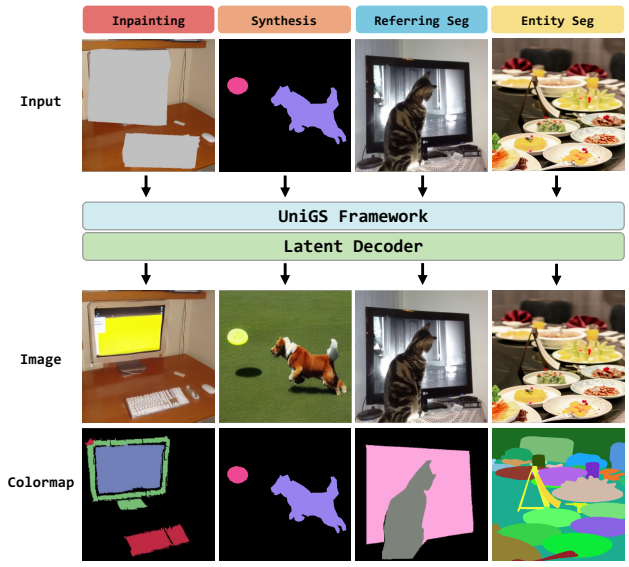

Figure 1. **Visualization results of a single UniGS model on image generation and segmentation.** We present four tasks: multi-class multi-region inpainting, image synthesis, referring segmentation, and entity segmentation. We note that the generation of colormaps shares a similar pipeline with images without needing any explicit segmentation loss.

1. Introduction

Deep learning has propelled the performance of several tasks to new heights, marking substantial progress within the computer vision community. Image generation [17, 27, 29, 74] and segmentation [6, 18, 32, 43, 45, 75], as two typical dense prediction tasks within this field, are widely used in plethora of applications such as autonomous driving [40], video surveillance [49], medical imaging [52], robotics [12], photography [56], and intelligent creation [61, 62].

The innovative usage of latent codes [46] in diffusion models has recently demonstrated remarkable capabilities in producing high-quality images, opening a new era of AI-generated content (AIGC). Nevertheless, using a similar design for segmentation remains relatively unexplored in diffusion-based works, despite evidence from specific studies [3, 5, 33, 36, 54] that highlight the potential of attention blocks to group pixels. Realizing such a capability with a unified representation for image and entity-level segmentation masks could potentially refine image generation, achieving greater coherence between the synthesized entities and their masks. Moreover, this unified representation offers significant potential for performing various dense prediction tasks, including both generation and segmentation in a single representation, as shown in Figure 1.

*Equal contribution. † indicates the corresponding author.

The intuitive solution is to represent segmentation masks simply as a colormap like Painter [59] and InstructDiffusion [16]. However, the implementation is far from straightforward for three main reasons. First, the colormap design should be consistent with latent space not explored in Painter [59]. Second, it should be able to differentiate entities in the same category. This challenge is not addressed in instructDiffusion [16], which can only detect one entity. Finally, the mask quality is not guaranteed and usually has many noises without regular segmentation loss functions like cross-entropy or dice loss. Even though the colormap design effectively achieves a unified representation, the large-scale dataset requirements for training diffusion models are at odds with the sparse segmentation annotations at hand, resulting in a critical bottleneck in our exploration.

To tackle these challenges, our first step is to validate that variational autoencoder (VAE) [23] used in stable diffusion [46] can effectively encode and decode colormaps in the same way as images. Based on colormap representation and latent diffusion model, we introduce the UniGS framework to simultaneously generate images and multi-class entity-level segmentation masks. The UniGS has a UNet architecture augmented with dual branches: one for image and another for mask generation. In the mask branch, we propose two modules, including a location-aware palette and a progressive dichotomy module. The former assigns each entity area to some fixed colors by the entities' center-of-mass location, enabling UniGS to discriminate entities within the same category. The latter efficiently decodes generated noisy colormap into explicit masks without knowing the entity numbers.

Then, we train our diffusion model under the inpainting protocol, addressing the scarcity of large-scale mask annotations. In this way, the diffusion model is primed to hone in on specific regions rather than the entire image. This flexibility facilitates using multiple segmentation datasets for training our diffusion model. Combining unified image and mask representation with an inpainting pipeline further integrates various tasks within a single representation with minor modifications. Figure 1 shows the effectiveness of the UniGS on four tasks, including multi-class multi-region inpainting, image synthesis, referring segmentation, and entity segmentation.

The main contributions of this work are as follows:

- We are the first to propose a unified diffusion model (UniGS) for image generation and segmentation within a unified representation by treating the entity-level segmentation masks the same as images.
- Two novel modules, including a location-aware palette and progressive dichotomy module, can make efficient transformations between the entity-level segmentation masks and colormap representations.
- The inpainting-based protocol addresses the scarcity of large-scale segmentation data and affords the versatility to employ a unified representation across multiple tasks.
- The extensive experiments show our UniGS framework's effectiveness on image generation and segmentation. In particular, UniGS can obtain segmentation performance comparable to state-of-the-art methods without any standard segmentation loss design. Our work can inspire foundation models with a unified representation for two mainstream dense prediction tasks.

2. Related Work

Diffusion Model for Generation. The diffusion models were initially introduced in the context of generation tasks [14] and have undergone significant evolution through latent design [46]. Diffusion models have been applied to a wide variety of generation [15, 39, 46, 50], image super-resolution [1, 13, 63], image inpainting [34, 53, 68, 71], image editing [22, 64, 73], image-to-image translation [11, 28, 57, 72], among others. We note that all current methods utilize the latent code to generate high-resolution images and have been extended to 3D [25, 37, 70] or video generation [4, 21, 35, 65]. Instead of those methods focusing on content generation, we endow the diffusion model with perception and segmentation ability by using similar representations for the images.

Diffusion Model for Segmentation. Several studies have delved into pixel-level segmentation [6, 18, 32, 41–45, 48, 75] using diffusion models through three distinct pipelines. The first two pipelines emphasize leveraging pre-trained stable diffusion [46] to simultaneously generate segmentation masks and images. Specifically, the first pipeline, as discussed in studies like [3, 5, 33, 36, 54], employs both self- and cross-attention maps in stable diffusion for shape grouping. However, these approaches demonstrate limited capabilities in the instance or entity-level discrimination [43, 45]. Conversely, the second pipeline [30, 66, 67, 69] primarily focuses on integrating a segmentation branch to produce precise mask generation by bringing substantially computational costs. Instead, the third pipeline [7, 8] is conditioned upon the input image by diffusing the image features to masks or bounding boxes. Furthermore, a prevalent issue with these methods is the inconsistency between image generation and segmentation mask generation processes. In contrast, we develop a unified representation for both tasks by converting the segmentation mask into a colormap.

Unified Representation. Some foundation models [16, 59, 60] explored unified representation for both generation and perception tasks. Our work is mostly similar to the Painter [59] and InstructDiffusion [16] but with various designs. Rather than reproducing the original color through MAE's [19] regression as in Painter [59], our approach in-

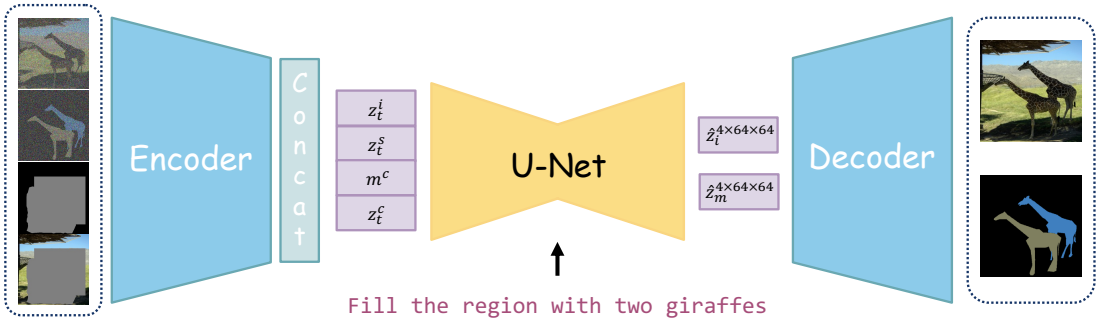

Figure 2. **Overview of the UniGS framework within the inpainting pipeline.** Similar to stable diffusion, our UniGS denoise the feature in the latent space by an encoder and decoder. We note that the predictions of UNet $\hat{z}_i$ and $\hat{z}_m$ are unified representations that can be decoded into images and colormaps by a similar latent decoder.

Notation	Definition	Notation	Definition
Υ	AutoEncoder (VAE)	Ω	Coarse Mask Generator
Ψ	Colormap Encoder	Φ	Colormap Decoder
M	Entity-level Masks	M_c	Colormap
I_0	Original Image	m_c	Coarse Mask

Table 1. **Illustration of some notations in the Method section.**

volves gradually diffusing the latent code of the colormap by several time steps. Compared to the InstructDiffusion [16], our framework offers greater flexibility in decoupling the image and colormap using distinct latent codes. As a result, there's no necessity to employ a lightweight segmentation branch for mask generation in our approach.

3. Method

Based on the latent diffusion model [46], the proposed UniGS framework aims to progressively and simultaneously denoise images and segmentation masks given a text prompt. In Figure 2, we show the overview of the UniGS model within the inpainting pipeline. Such a pipeline can address the challenge of insufficient segmentation datasets and unifying multiple tasks in a single representation.

Specifically, the input of our UNet has four parts, including the latent encode of the noised image, colormap, context, and a resized coarse mask. They are denoted by z_t^i, z_t^s, z_t^c and m^c, respectively. Based on the text prompt, we use an UNet to denoise the z_t^i, z_t^s to $\hat{z}_i$ and $\hat{z}_m$. During inference, z_t^i and z_t^s would be the pure Gaussian noise. Compared to stable diffusion [46], there is no obvious structure difference except for the input and output channel numbers.

In the following, we begin with an overview of latent diffusion techniques for high-resolution image synthesis. Then, we introduce our novel mask representation to represent entity masks. Lastly, we propose our whole inpainting pipeline and its extension to multiple tasks. It is noted that Table 1 lists essential notions in this section.

3.1. Review of Latent Diffusion

Diffusion models [20] is a class of likelihood-based models that define a Markov chain of forward and backward processes, gradually adding and removing noise to sample data. The forward process is defined as

$$q(z_t|z_0) = \mathcal{N}(z_t|\sqrt{\bar{\alpha}_t}z_0, (1-\bar{\alpha}_t)z_0), \tag{1}$$

which transforms data sample z_0 to a latent noisy sample z_t for $t \in \{0, 1, ..., T\}$ by adding gaussian noise ϵ to z_0. $\bar{\alpha}_t := \prod_{s=0}^{t} \alpha_s = \prod_{s=0}^{t}(1-\beta_s)$ where β_s represents the noise variance schedule [20]. During training, a neural network (usually an UNet) $f_\theta(\boldsymbol{z}_t, t)$ is trained to predict ϵ to recover z_0 from z_t by minimizing the training objective with ℓ_2 loss [20]:

$$\mathcal{L}_{\text{train}} = \frac{1}{2}||f_\theta(z_t, t) - \epsilon||^2. \tag{2}$$

where θ is the parameters of the neural network. At inference stage, data sample z_0 is reconstructed from z_T with the model f_θ and an updating rule [20, 51] in an iterative way, *i.e.*, $z_T \rightarrow z_{T-\Delta} \rightarrow ... \rightarrow z_0$. For a clear illustration, we omit the updating rule and regard the output of $f_\theta(z_t, t)$ as z_0. In the context of generating high-resolution images $I_0 \in \mathcal{R}^{h\times w\times 3}$ given a text prompt, diffusion models would incur substantial computational costs if using large image size like $h = w = 512$. The h and w represent the image height and width. To tackle this issue, latent diffusion models (LDM) uses latent code of the images as $z_0^i \in \mathcal{R}^{\frac{h}{4}\times\frac{w}{4}\times 4}$ [58]:

$$z_0^i = \Upsilon(I) \quad \text{and} \quad \hat{I}_0 = \Upsilon^{-1}(\hat{z}_0^i) \tag{3}$$

where Υ and Υ^{-1} represent the encoding and decoding process of AutoEncoder (VAE) to I. $\hat{*}$ indicates the prediction results. As such, latent diffusion reduces computational demands and maintains good generation ability. We base our UniGS model on Stable Diffusion [46], a popular LDM variant.

3.2. Colormap-based Entity Mask Representation

We use a colormap representing segmentation masks that can align mask representation with image format while supporting variability in entity numbers. However, designing a colormap encoder and decoder is non-trivial due to the requirements of discriminating each entity within the same categories. Moreover, this representation would lack the standard segmentation loss in latent space like binary cross-entropy and dice loss. Using the denoise loss in Eq 2 for colormap would lead to several extreme cases in Figure 3. Thus, we describe our location-aware palette and progressive dichotomy modules in colormap encoding and decoding to solve the above-mentioned problems.

Colormap Encoding. The colormap encoder Ψ converts several entity-level binary segmentation masks $M \in \{0,1\}^{n\times h\times w}$ to an colormap $M_c \in [0,255]^{h\times w\times 3}$ as

$$M_c = \Psi(M) \tag{4}$$

n denotes the number of sampled entities. The M_c is initialized by zero value and then assigned some color for each entity area by our location-aware palette. Specifically, we partition an image into $b \times b$ grids where each grid has a unique color. Each entity area is associated with these fixed colors if their gravity centers are at the grids. Each RGB channel has five candidate color values $\{0, 64, 128, 192, 255\}$ in our location-aware palette. Thus, the overall color number is $124 = 5^3 - 1$ with color $(0, 0, 0)$ indicating the background. The grid numbers $b^2 = |b \times b|$ should be less than 124.

The location-aware palette design proves simple but efficient in covering nearly all labeled entities (97.4% coverage ratio across the COCO, ADE20K, OpenImages, and EntitySeg datasets). That's because UNet has a position encoding design that can help predict corresponding colors. In contrast, random color assignments often struggle to distinguish between entities of the same category due to providing too large a color space.

Colormap Decoding. While the generated colormap effectively differentiates between entities visually, converting it into the perfect entity-level masks presents several challenges. A primary issue is the need for more awareness of entity numbers. Therefore, heuristic k-means clustering is impractical. To tackle this issue, we propose a progressive dichotomy module Φ to group areas of identical color by pixel-level features p without prior knowledge of cluster numbers.

$$\hat{M} = \Phi(\hat{M}_c) = \Phi(\Upsilon^{-1}(\hat{z}_0^s)) \tag{5}$$

where $\hat{M}_c$ is predicted colormap decoded by VAE, and $\hat{M} \in \{0,1\}^{n\times H\times W}$ has n binary masks.

Specifically, the progressive dichotomy module (PDM) is a depth-first cascaded clustering method where we further split the j^{th} entity mask $\hat{m}_{v-1}^j$ generated at $(v-1)^{\text{th}}$ iteration into the two sub-masks $\hat{m}_v^{2j}$ and $\hat{m}_v^{2j+1}$ at v^{th} iteration,

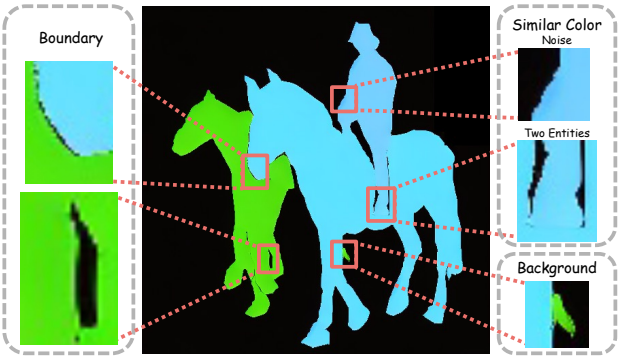

Figure 3. **Illustration of several difficult cases in the decoded color map.** We conclude those cases into three kinds of problems, including boundary, similar color, and background.

$$\{\hat{m}_v^{2j}, \hat{m}_v^{2j+1}\} = \mathcal{BK}(\hat{m}_{v-1}^j) \tag{6}$$

The $\mathcal{BK}$ denotes two-cluster k-means and each $\hat{m} \in \{0,1\}^{H\times W}$. Further splitting $\hat{m}_{v-1}^j$ will stop until the average L2 distance of mask pixels to their mean less than δ:

$$\frac{\sum_{o\in\hat{m}_{v-1}^j}(p_o - c_{\hat{m}_{v-1}^j})^2}{|\hat{m}_{v-1}^j|} < \delta \tag{7}$$

The $c_{\hat{m}_{v-1}^j} = \frac{\sum_{o\in\hat{m}_{v-1}^j} p_o}{|\hat{m}_{v-1}^j|}$ with $|\hat{m}_{v-1}^j|$ denoting the pixel numbers of mask $\hat{m}_{v-1}^j$.

The pixel feature p_o is designed in light of three critical observations in Figure 3. $o \in [0, ..., h \times w)$. At first, it is not trivial to discern whether a gradient color signifies one or multiple entities. Second, the foreground colors would be degraded by the background. Thirdly, some black holes are hard to predict as true or false positives. Thus, we design $p_o \in \mathcal{R}^{1\times 6}$ with both RGB and LAB image space. Including LAB image space is pivotal due to their perceptual uniformity property, which ensures that minor variations in LAB values translate to approximately uniform alterations in color as perceived by the human eye, thereby providing enhanced contrast.

3.3. Inpainting Pipeline

We adopt an inpainting pipeline for training and inference to reconcile the generative model's requirements for large-scale segmentation datasets. For example, the Open-Images dataset [2] with mask annotations encompasses approximately 1.8 million images but only contains about three entity-level labels per image. Directly training the latent diffusion model results in too many ambiguities due to unlabeled areas. Instead, our inpainting pipeline enables the

Task	Condition			Output	
	coarse mask (m^c)	control factor (z_i^c)	text prompt template	image ($\hat{z}_0^i$)	mask ($\hat{z}_0^s$)
Inpainting	$\Omega(M)$	$\Upsilon(I_0 \odot (1 - m^c))$	'inpainting: generate dog.'	✓	✓
Image Synthesis	$\mathcal{J}$	$\Upsilon(M_c)$	'synthesis: generate dog, ground, and sky.'	✓	✗
Referring Segmentation	$\Omega(M)$	$\Upsilon(I_0)$	'referring: find dog.'	✗	✓
Entity Segmentation	$\mathcal{J}$	$\Upsilon(I_0)$	'panoptic: all entities.'	✗	✓

Table 2. **Illustration of the condition signal's design on training task in our framework.** The ✓and ✗indicate whether we expect the two output tensors to be the same as our condition z_i^c.

generative model to concentrate on the valid areas regardless of the partial segmentation labels.

In the training period, the UNet input is $z^u \in \mathcal{R}^{\frac{H}{4}\times\frac{W}{4}\times 13}$ that concatenated by

$$z_t^u = \text{CONCAT}(z_t^i, z_t^s, m^c, z_t^c) \tag{8}$$

z_t^i and z_t^s are the latent code of the noised image I_t and colormap S_t in time step t where both z_t^i and z_t^s are $\mathcal{R}^{\frac{h}{4}\times\frac{w}{4}\times 4}$. $m^c \in \{0,1\}^{\frac{h}{4}\times\frac{w}{4}\times 1}$ is a coarse mask where 1 indicates a rectangular or an irregular area that needs our UniGS framework to fill entities and their masks,

$$m_c = \Omega(M) \tag{9}$$

More details regarding Ω are available in the supplementary material. Next, z_i^c is the latent code of the masked image by m_c,

$$z_t^c = \Upsilon(I_0 \odot (1 - m^c)) \tag{10}$$

The UNet output is

$$\hat{z}_0 = f_\theta(z_t^u, t) \in \mathcal{R}^{\frac{H}{4}\times\frac{W}{4}\times 8} \tag{11}$$

where the first and last four channels of $\hat{z}_0$ can be latently decoded to the final image and colormaps.

3.4. One-to-Many Tasks

The inpainting pipeline with colormap representation allows for integrating various tasks within a single model. We use the UniGS model for four vision tasks: inpainting, image synthesis, referring segmentation, and entity segmentation. The configuration of each task is presented in Table 2.

Multi-class Multi-region Inpainting. Our baseline task that has been detailed in Section 3.3.

Image Synthesis: z_i^c is latent code of colormap M_c containing sampled entities. Meanwhile, the coarse mask m_c is $\mathcal{J}$ matrix of all ones to cover the entire image area. For the output, we maintain $\hat{z}_0^s = \Upsilon(M_c)$ and predict the entities' appearance $\hat{z}_0^i$.

Referring Segmentation. This task aims at segmenting some classes based on instructions. Thus, we preserve image information by $z_i^c = \Upsilon(I_0)$. Considering the requirement of negative samples to ensure alignment between the entity and text prompt, we define λ as the possibility of each sampled entity belonging to a negative in training. For negative samples, the category names in text prompts are replaced with others that do not appear in the coarse mask.

Entity Segmentation. All the entities should be predicted in $\hat{z}_0^s$ with the coarse mask area $\mathcal{J}$.

4. Experiments

In this section, we first explore the performance of our proposed UniGS in four individual tasks, including multi-class multi-region inpainting, image synthesis, referring segmentation, and entity segmentation. Some key module designs or hyper-parameters on the mask quality are ablated in referring segmentation. Similar to other works [9, 45, 47] to evaluate the image and mask quality, we use the intersection over union (IoU) and recall for mask evaluation and the Fréchet inception distance (FID) and CLIP score (CS) for image generation.

4.1. Experiment Setting

For each single-task model, we exclusively utilize the COCO dataset [31], the Open Images [26], and EntitySeg datasets [45] as our training data. Considering the COCO panoptic data having about 10% ignored area, we only use the EntitySeg for entity segmentation task in case of performance degradation.

In our training process, we randomly sample up to four objects per sampled area for tasks such as inpainting, image synthesis, and referring segmentation. On the other hand, entity segmentation should include all the entities that can cover the whole sampled area. During the inference period, we sample 1000 images in COCO validation data as our test set, where each image has a coarse mask and various control factors for different tasks, as shown in Table 2.

We initialize our model with stable diffusion v1.5 inpainting and weight newly added channels as zero. The image size and latent factor reduction ratio are set to 512×512.

4.2. Multi-class Multi-region Inpainting

We evaluate the inpainting model by inserting one or multiple objects into the coarse mask area generated from the entity masks. The model's output in this task includes the

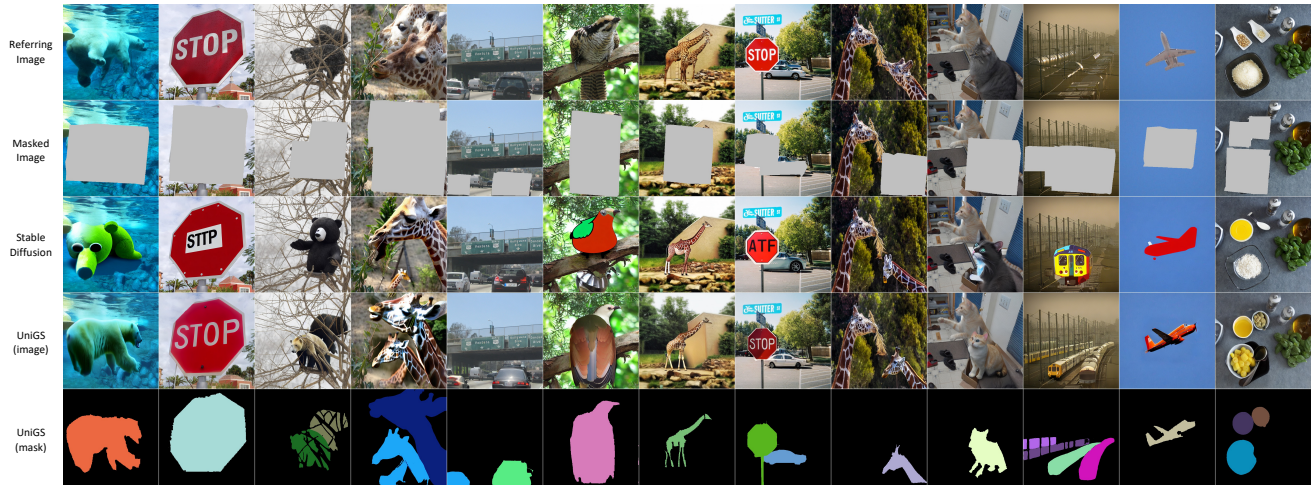

Figure 4. **Qualitative comparison of inpainting results between Stable Diffusion and our UniGS.** For the coarse masks, we keep the consistency of the ones used in our training phase to eliminate the pattern gap. Furthermore, we showcase multi-class, multi-region inpainting to the multiple entities within the same category, moving beyond the conventional approach of incorporating a single entity.

Method	FID (↓)	CLIP Score (↑)	FID (↓)	CLIP Score (↑)
	single object		multiple objects	
$SD_I^{1.5}$	4.95	88.86	7.82	83.80
$UniGS^{1.4}$	4.39	88.92	6.19	84.43
$UniGS_I^{1.5}$	**3.78**	**90.22**	**5.89**	**85.87**

Table 3. **Quantitative results on image inpainting task.** The $SD_I^{1.5}$ means the stable diffusion inpainting model with version 1.5. The $UniGS^{1.4}$ is the UniGS that initialized from the stable diffusion model with version 1.4, the $UniGS_I^{1.5}$ is the UniGS that initialized from stable diffusion inpainting model with version 1.5.

generated image and colormap. As in Table 3, our method outperformed the original model regarding both FID and CLIP scores. This improvement was observed even when our model was initialized using the stable diffusion 1.4 pre-trained model, highlighting the effectiveness of our approach in enhancing image generation through the integration of object mask guidance. That's because our unified representation effectively constrains the model to maintain consistency between the visual appearance of the objects and their corresponding masks. As a result, the object masks impart a robust shape priority, guiding and refining the image generation process to ensure alignment and coherence in the final output.

Figure 4 shows the visual comparison between stable diffusion and our UniGS with the coarse masks generated from our code used in the training period. In other words, our testing keeps a similar pattern of inpainting area. It is evident from these results that the objects generated by the model are in strong harmony with the high-quality masks, showcasing the model's effectiveness in seamlessly integrating the objects into the overall image composition. Furthermore, this impressive coherence between generated objects and their masks is attributable to our model's unified representation of images and segmentation masks.

4.3. Image Synthesis

In this task, we expect to take a colormap as input along with a text-based image synthesis prompt and output a synthesized image. Except for the conventional metrics of Fréchet inception distance (FID) and CLIP score, we incorporated mean Intersection over Union (mIoU) to evaluate the alignment of the generated image with the specified mask shape. For this external evaluation, we utilized the Mask2Former model equipped with a large swan backbone to perform segmentation on the images generated by our model. The mIoU is then calculated by comparing these segmentation masks against the original colormap. In Figure 5, we present the visual consistency between the synthesized objects and the provided color masks among four methods: stable diffusion, ControlNet, T2I Adaptor, and UniGS. We modify the pipeline of the compared approaches for a fair comparison. For stable diffusion designed not for image synthesis, we only use text prompts for conditions. For ControlNet and T2I-Adaptor, we follow the default settings and input the segmentation map and text prompt to get the synthesis image. In this figure, we highlight the model's ability to align the generated objects with the specified mask constraints closely. Moreover, those visualization results reflect the successful and seamless integration of these objects within their backgrounds, further highlighting the benefits of our unified representation in synthesizing contextually coherent and visually harmonious images. As shown in Table 4, our method is more favorable in all critical metrics,

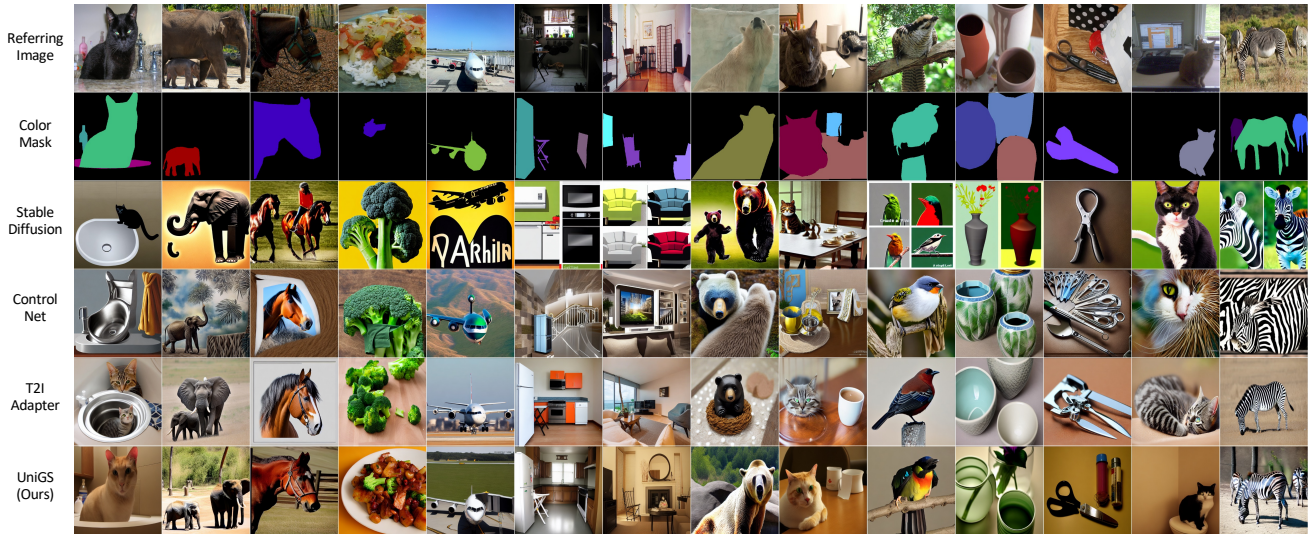

Figure 5. **Qualitative comparison among Stable Diffusion, ControlNet, T2I-Adapter, and our UniGS on the image synthesis task.** Compared to those methods, UniGS maintains more coherence between the generated entities and their corresponding segmentation masks.

Method	FID (↓)	CS (↑)	mIoU (↑)	FID (↓)	CS (↑)	mIoU (↑)
	single object			multiple object		
SD [70]	36.502	54.708	0.196	34.770	56.511	0.191
CN [72]	35.111	55.230	0.277	30.108	58.709	0.326
T2I [38]	34.434	59.024	0.306	24.898	62.910	0.379
UniGS	**15.272**	**65.015**	**0.781**	**14.271**	**69.504**	**0.777**

Table 4. **Quantitative results on image synthesis task.** 'SD', 'CN', and T2I indicate the stable diffusion, CrontrolNet, and T2I-Adaptor. The 'CS' is the CLIP Score.

Method	Backbone	mIoU(↑)	Recall (↑)
Mask2Former [10]	Swin-Large	**0.815**	**0.887**
UniGS (Ours)	-	0.808	0.872

Table 5. **Quantitative results on referring segmentation task.** We choose Mask2Former with a Swin-Large backbone as our baseline for comparison with SOTA segmentation methods.

including the FID, CLIP score, and mIoU. This comparison underscores that the unified representation for both image and segmentation mask can help the image synthesis network have a higher quality perceptual judgment and more precise mask-to-object alignment.

4.4. Referring Segmentation

We evaluate the quality of generated masks by mIoU and recall metrics for the referring segmentation task. In Table 5 on the comparison to the state-of-the-art segmentation method Mask2Former [10], our generative method has considerable segmentation quality. These results are worth noticing as we do not use any explicit segmentation loss, thereby demonstrating the potential of the UniGS model.

Method	Backbone	mIoU(↑)	Recall (↑)	AP^e (↑)
ConInst-Entity [43, 55]	Swin-Large	0.621	0.685	0.397
SAM [24]	VIT-Huge	0.653	0.714	0.432
CropFormer [45]	Swin-Large	**0.664**	**0.727**	**0.449**
UniGS (Ours)	-	0.631	0.692	0.407

Table 6. **Quantitative results on entity segmentation task**. The AP^e is AP with a non-overlapped constraint used in entity segmentation.

Method	mIoU(↑)	Recall (↑)
Random Color Assignment	0.493	0.563
Location-aware Palette (Ours)	**0.808**	**0.872**

Table 7. **Ablation study on various color assignments in mask encoder.** The 'Random Color Assignment' indicates assigning each entity with a random color.

4.5. Entity Segmentation

The entity segmentation aims at splitting an input image into several semantically coherent regions. Thus, the generated colormap should cover the whole image. After latent decoding the output from UNet, we use the progressive dichotomy module to transform the colormap into explicit segmentation masks. In Table 6, we show that there is still a significant performance gap between the UniGS and state-of-the-art entity segmentation model. However, we mention that the entity performance of the UniGS model is acceptable and better than kernel-based methods like CondInst.

4.6. Unified Model

Our model cannot only perform a single task as a single model but also achieve multi-tasking as a unified model that text prompts can differentiate. Furthermore, we add a group of learnable task embeddings to indicate each task better.

This task embedding would be added to the position embedding. In Table 8, the unified model performs best on the four tasks with enough training epochs. That means four tasks can benefit each other in a unified representation.

Method	Epoch	Inpainting Synthesis	Referring Seg	Entity Seg
		FID(↓)	mIoU(↑)	mIoU(↑)
Single Model	48	5.890	14.271	0.808	0.631
Unified Model	48	6.312	23.187	0.798	0.604
Single Model	200	5.886	14.354	0.810	0.626
Unified Model	200	**5.494**	**13.920**	**0.838**	**0.649**

Table 8. **The comparison between each single best and unified model on all four tasks.** "Unified model" is trained in multi-tasks, and "Single model" is selected as the best model for each single task.

4.7. Ablation Study

In the following, we ablate different color assignment criteria and progressive dichotomy modules with various hyperparameters. All the ablation studies are conducted in referring segmentation tasks to measure the mask quality.

Location-aware Palette. To evaluate the effectiveness of our color mapping over the random color assignment for the object, we have individually trained the referring segmentation models for both color mapping methods, as shown in Table 7. Our color mapping method lets the model easily learn the pattern of the object color mask.

Progressive Dichotomy Module. Compared to the fixed cluster numbers in k-means, our proposed progressive dichotomy module has the advantage of adaptive cluster numbers. We verify our method in Table 9 by comparing K-Means and ours. Our progressive dichotomy module has no noticeable performance degradation compared to K-Means, even with knowing the ground truth numbers, manifesting the effectiveness and robustness of our progressive dichotomy module.

Furthermore, the distance threshold δ and pixel feature used in the progressive dichotomy module are ablated in Table 10. In Table 10(a), we can see that the distance threshold designed in the progressive dichotomy module is robust to the segmentation performance ranging from 0 to 20. In Table 10(b), using the RGB and LAB space pixel feature to decode the generated colormap can obtain the best mask quality because the LAB space can offer more contrast information for those two similar colors in RGB space.

Table 11 illustrates a comparison of our PDM to other clustering methods. We note that the agglomerative method requires large memory, leading to out-of-memory in our experiments.

5. Conclusion

This paper introduces a novel, effective, unified representation of image generation and segmentation tasks. The key to our approach is regarding entity-level segmentation masks as a colormap generation problem. To distinguish entities within the same category, we employ a location-aware palette where each entity is distinctly colored based on its center-of-mass location. Furthermore, our progressive dichotomy module can efficiently transform a generated, albeit noisy, colormap into high-quality segmentation masks. Our extensive experiments on four diverse tasks demonstrate the robustness and versatility of our unified representation in image generation and segmentation. In the future, we will explore the multi-task training of our unified representation in a single model. We hope our work can foster the development of a foundation model with a unified representation for various tasks.

Method	Cluster Numbers	mIoU(↑)	Recall (↑)
Native K-Means	Fixed (3)	0.520	0.641
	Adaptive (GT)	**0.810**	**0.874**
PDM	Adaptive	0.808	0.872

Table 9. **Comparison between native K-Means and our progressive dichotomy module.** 'Fixed (3)' indicates that we assign native K-Means with three cluster numbers. 'Adaptive (GT)' is to assign the cluster number by the ground truth number.

δ	mIoU(↑)	Recall (↑)
1	0.804	0.868
10	**0.808**	**0.872**
20	0.791	0.857
50	0.705	0.789

(a)

pixel feature	mIoU(↑)	Recall (↑)
RGB	0.796	0.860
LAB	0.787	0.856
RGB + LAB	**0.808**	**0.872**

(b)

Table 10. **Ablation study on progressive dichotomy module.** We ablate the distance threshold δ (a) and pixel feature (b) in the colormap decoding process.

Method	Cluster Numbers	Time (s)	Memory (MB)	mIoU(↑)	Recall (↑)
Native K-Means	Fixed (3)	0.51	240	0.520	0.641
	Adaptive (GT)	0.58	279	**0.810**	**0.874**
DBSCAN	Adaptive	14.20	677	0.341	0.398
Agglomerative	Adaptive	Inf	OOM	-	-
PDM	Adaptive	1.47	295	0.808	0.872

Table 11. **The comparison to other clustering methods.** The 'PDM' indicates the proposed progressive dichotomy module. The native k-means, DBSCAN, and agglomerative methods are the standard clustering methods.

6. Acknowledgement

Supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DOI/IBC) contract number 140D0423C0074. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government.

References

[1] Georgios Batzolis, Jan Stanczuk, Carola-Bibiane Schönlieb, and Christian Etmann. Conditional image generation with score-based diffusion models. *arXiv preprint arXiv:2111.13606*, 2021. 2

[2] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-scale interactive object segmentation with human annotators. In *CVPR*, 2019. 4

[3] Ryan Burgert, Kanchana Ranasinghe, Xiang Li, and Michael S Ryoo. Peekaboo: Text to image diffusion models are zero-shot segmentors. *arXiv preprint arXiv:2211.13224*, 2022. 1, 2

[4] Duygu Ceylan, Chun-Hao P Huang, and Niloy J Mitra. Pix2video: Video editing using image diffusion. In *ICCV*, 2023. 2

[5] Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *arXiv preprint arXiv:2301.13826*, 2023. 1, 2

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2017. 1, 2

[7] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. In *ICCV*, 2023. 2

[8] Ting Chen, Lala Li, Saurabh Saxena, Geoffrey Hinton, and David J Fleet. A generalist framework for panoptic segmentation of images and videos. In *ICCV*, 2023. 2

[9] Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. *arXiv preprint arXiv:2307.09481*, 2023. 5

[10] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, pages 1290–1299, 2022. 7

[11] Bin Cheng, Zuhao Liu, Yunbo Peng, and Yue Lin. General image-to-image translation with one-shot image guidance. In *ICCV*, 2023. 2

[12] Ruihang Chu, Yukang Chen, Tao Kong, Lu Qi, and Lei Li. Icm-3d: Instantiated category modeling for 3d instance segmentation. *RAL*, 2021. 1

[13] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In *CVPR*, 2022. 2

[14] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021. 2

[15] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-based generative modeling with critically-damped langevin diffusion. In *ICLR*, 2022. 2

[16] Zigang Geng, Binxin Yang, Tiankai Hang, Chen Li, Shuyang Gu, Ting Zhang, Jianmin Bao, Zheng Zhang, Han Hu, Dong Chen, et al. Instructdiffusion: A generalist modeling interface for vision tasks. *arXiv preprint arXiv:2309.03895*, 2023. 2, 3

[17] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. In *ICML*, 2015. 1

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1, 2

[19] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 2

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 3

[21] 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. 2

[22] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *CVPR*, 2023. 2

[23] Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. 2

[24] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *ICCV*, 2023. 7

[25] Juil Koo, Seungwoo Yoo, Minh Hieu Nguyen, and Minhyuk Sung. Salad: Part-level latent diffusion for 3d shape generation and manipulation. In *ICCV*, 2023. 2

[26] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020. 5

[27] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip Torr. Controllable text-to-image generation. *NeurIPS*, 2019. 1

[28] Bo Li, Kaitao Xue, Bin Liu, and Yu-Kun Lai. Bbdm: Image-to-image translation with brownian bridge diffusion models. In *CVPR*, 2023. 2

[29] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. Mat: Mask-aware transformer for large hole image inpainting. In *CVPR*, 2022. 1

[30] Ziyi Li, Qinye Zhou, Xiaoyun Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. Guiding text-to-image diffusion model towards grounded generation. In *ICCV*, 2023. 2

[31] 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 *ECCV*, pages 740–755. Springer, 2014. 5

[32] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, 2018. 1, 2

[33] Xuyang Liu, Siteng Huang, Yachen Kang, Honggang Chen, and Donglin Wang. Vgdiffzero: Text-to-image diffusion models can be zero-shot visual grounders. *arXiv preprint arXiv:2309.01141*, 2023. 1, 2

[34] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022. 2

[35] Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. Videofusion: Decomposed diffusion models for high-quality video generation. In *CVPR*, 2023. 2

[36] Chaofan Ma, Yuhuan Yang, Chen Ju, Fei Zhang, Jinxiang Liu, Yu Wang, Ya Zhang, and Yanfeng Wang. Diffusionseg: Adapting diffusion towards unsupervised object discovery. *arXiv preprint arXiv:2303.09813*, 2023. 1, 2

[37] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *CVPR*, 2023. 2

[38] Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023. 7

[39] 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. In *ICML*, 2021. 2

[40] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Amodal instance segmentation with kins dataset. In *CVPR*, 2019. 1

[41] Lu Qi, Jason Kuen, Jiuxiang Gu, Zhe Lin, Yi Wang, Yukang Chen, Yanwei Li, and Jiaya Jia. Multi-scale aligned distillation for low-resolution detection. In *CVPR*, 2021. 2

[42] Lu Qi, Yi Wang, Yukang Chen, Ying-Cong Chen, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Pointins: Point-based instance segmentation. *TPAMI*, 2021.

[43] Lu Qi, Jason Kuen, Yi Wang, Jiuxiang Gu, Hengshuang Zhao, Philip Torr, Zhe Lin, and Jiaya Jia. Open world entity segmentation. *TAPMI*, 2022. 1, 2, 7

[44] Lu Qi, Jason Kuen, Weidong Guo, Jiuxiang Gu, Zhe Lin, Bo Du, Yu Xu, and Ming-Hsuan Yang. Aims: All-inclusive multi-level segmentation for anything. In *NeurIPS*, 2023.

[45] Lu Qi, Jason Kuen, Tiancheng Shen, Jiuxiang Gu, Wenbo Li, Weidong Guo, Jiaya Jia, Zhe Lin, and Ming-Hsuan Yang. High quality entity segmentation. In *ICCV*, 2023. 1, 2, 5, 7

[46] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1, 2, 3

[47] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, 2023. 5

[48] Tiancheng Shen, Yuechen Zhang, Lu Qi, Jason Kuen, Xingyu Xie, Jianlong Wu, Zhe Lin, and Jiaya Jia. High quality segmentation for ultra high-resolution images. In *CVPR*, 2022. 2

[49] Guang Shu. Human detection, tracking and segmentation in surveillance video. 2014. 1

[50] Abhishek Sinha, Jiaming Song, Chenlin Meng, and Stefano Ermon. D2c: Diffusion-decoding models for few-shot conditional generation. In *NeurIPS*, 2021. 2

[51] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 3

[52] Paul Suetens. *Fundamentals of medical imaging*. Cambridge university press, 2017. 1

[53] David Svitov, Dmitrii Gudkov, Renat Bashirov, and Victor Lempitsky. Dinar: Diffusion inpainting of neural textures for one-shot human avatars. In *ICCV*, 2023. 2

[54] Junjiao Tian, Lavisha Aggarwal, Andrea Colaco, Zsolt Kira, and Mar Gonzalez-Franco. Diffuse, attend, and segment: Unsupervised zero-shot segmentation using stable diffusion. *arXiv preprint arXiv:2308.12469*, 2023. 1, 2

[55] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *ECCV*, 2020. 7

[56] Yu-Ju Tsai, Yu-Lun Liu, Lu Qi, Kelvin CK Chan, and Ming-Hsuan Yang. Dual associated encoder for face restoration. *arXiv preprint arXiv:2308.07314*, 2023. 1

[57] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*, 2023. 2

[58] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *NeurIPS*, 2017. 3

[59] Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *CVPR*, 2023. 2

[60] Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. Seggpt: Segmenting everything in context. In *ICCV*, 2023. 2

[61] Yi Wang, Lu Qi, Ying-Cong Chen, Xiangyu Zhang, and Jiaya Jia. Image synthesis via semantic composition. In *ICCV*, 2021. 1

[62] Yi Wang, Menghan Xia, Lu Qi, Jing Shao, and Yu Qiao. Palgan: Image colorization with palette generative adversarial networks. In *ECCV*, 2022. 1

[63] Chanyue Wu, Dong Wang, Yunpeng Bai, Hanyu Mao, Ying Li, and Qiang Shen. Hsr-diff: hyperspectral image super-resolution via conditional diffusion models. In *ICCV*, 2023. 2

[64] Chen Henry Wu and Fernando De la Torre. A latent space of stochastic diffusion models for zero-shot image editing and guidance. In *ICCV*, 2023. 2

[65] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *ICCV*, 2023. 2

[66] 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. In *NeurIPS*, 2023. 2

[67] 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. In *ICCV*, 2023. 2

[68] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *CVPR*, 2023. 2

[69] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *CVPR*, 2023. 2

[70] Minkai Xu, Alexander S Powers, Ron O Dror, Stefano Ermon, and Jure Leskovec. Geometric latent diffusion models for 3d molecule generation. In *ICML*, 2023. 2, 7

[71] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *CVPR*, 2023. 2

[72] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 2, 7

[73] Zhixing Zhang, Ligong Han, Arnab Ghosh, Dimitris N Metaxas, and Jian Ren. Sine: Single image editing with text-to-image diffusion models. In *CVPR*, 2023. 2

[74] Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout. In *CVPR*, 2019. 1

[75] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 1, 2