# A Simple Latent Diffusion Approach for Panoptic Segmentation and Mask Inpainting

Wouter Van Gansbeke⋆ and Bert De Brabandere[1]

[1] Segments.ai

**Abstract.** Panoptic and instance segmentation networks are often trained with specialized object detection modules, complex loss functions, and ad-hoc post-processing steps to manage the permutation-invariance of the instance masks. This work builds upon Stable Diffusion and proposes a latent diffusion approach for panoptic segmentation, resulting in a simple architecture that omits these complexities. Our training consists of two steps: (1) training a shallow autoencoder to project the segmentation masks to latent space; (2) training a diffusion model to allow image-conditioned sampling in latent space. This generative approach unlocks the exploration of mask completion or inpainting. The experimental validation on COCO and ADE20k yields strong segmentation results. Finally, we demonstrate our model's adaptability to multi-tasking by introducing learnable task embeddings. The code and models will be made available.[1]

**Keywords:** Latent Diffusion · Panoptic Segmentation · Dense Prediction

## 1 Introduction

The image segmentation task [45, 49] has gained popularity in the literature, encompassing three popular subfields: semantic, instance, and panoptic segmentation. Over the years, segmentation tools have proven their usefulness for a wide range of applications, such as autonomous driving [18], medical imaging [48], agriculture [16], and augmented reality [1, 24]. Current methods are built upon convolutional networks [28] and transformers [20, 70] to learn hierarchical image representations, and to simultaneously leverage large-scale datasets [43, 84]. Earlier segmentation approaches relied on specialized architectures [7, 27, 65, 74], such as region proposal networks [57] and dynamic convolutions [33]. More recent approaches [8, 14] advocate for an end-to-end strategy but introduce complex loss functions, *e.g.*, bipartite matching. Some works have shown promising results without the necessity for labels [17, 67–69, 73, 75], but likewise require highly-specialized modules, such as region proposal networks or clustering. Differently, we seek to leverage generative models to bypass the aforementioned components. This observation aligns with recent works [9–11, 40, 47] advocating for general computer vision models in an attempt to unify the field. These pioneering works

---

⋆ This work was done while the author was at Segments.ai, and the final training run was conducted at INSAIT. The author is now affiliated with Google DeepMind.
[1] https://github.com/segments-ai/latent-diffusion-segmentation
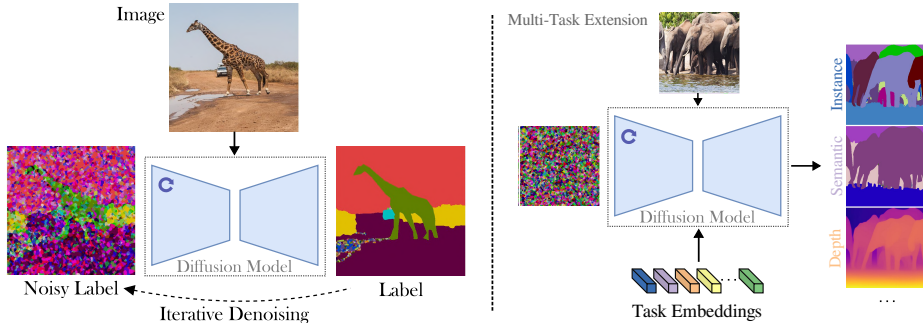
**Fig. 1:** (*Left:*) We present a simple generative approach for panoptic segmentation that builds upon Stable Diffusion [58]. The key idea is to leverage the diffusion process to bypass complex detection modules and to unlock mask inpainting. The generative process is conditioned on RGB images to iteratively predict the masks. (*Right:*) Our framework can be extended to a multi-task setting by introducing task embeddings.

limit the adoption of task-specific components but instead use a generative process. In a similar vein, we take inspiration from recent text-to-image diffusion models [19, 29, 30, 50, 51, 55, 56, 58] to tackle the segmentation task in a generative fashion. In addition to the diffusion model's general architecture, we further motivate this decision as follows: (*i*) diffusion models generate images with high photorealism and diversity; (*ii*) perform on par with autoregressive priors while being more computationally efficient [55, 62]; (*iii*) learn high-quality spatial representations, advantageous for dense prediction tasks; (*iv*) naturally exhibit image-editing capabilities. We now aim to build upon these properties.

To realize this objective, we introduce **LDMSeg**, a simple **L**atent **D**iffusion **M**odel for **Seg**mentation, visualized in Figure 1. Our contributions are fourfold:

- **Generative Framework:** We propose a fully generative approach based on Latent Diffusion Models (LDMs) for panoptic segmentation. We build upon Stable Diffusion [58] to strive for simplicity and computational efficiency.
- **General-Purpose Design:** As a result, we circumvent specialized architectures, complex loss functions, and object detection modules, present in the majority of prevailing methods. Here, the denoising objective omits the necessity for object queries, region proposals, and Hungarian matching [41] to handle the permutation-invariance of the instances.
- **Mask Inpainting:** We apply our method to scene-centric datasets. In contrast to prior art, we demonstrate mask inpainting for different sparsities.
- **Multi-Task Framework:** Our simple and general approach can easily be extended to train a single generative model for multiple tasks, like instance segmentation, semantic segmentation and depth prediction. Querying the model for a different task merely requires changing the task embedding.

To the best of our knowledge, this paper presents the first latent diffusion approach that achieves strong results for panoptic segmentation, while also extending to mask inpainting and multi-task learning.

## 2   Related Work

**Panoptic Segmentation.** Panoptic segmentation [37] has lately gained popularity as it combines semantic and instance segmentation. In particular, its goal is to detect and segment both *stuff-like* (*e.g.*, vegetation, sky, mountains, *etc.*) and *thing-like* (*e.g.*, person, cat, car, *etc.*) categories. Earlier works modified instance segmentation architectures to additionally handle (amorphous) *stuff* categories as they are hard to capture with bounding boxes. For instance, Kirillov *et al.* showed promising results by making independent predictions using semantic and instance segmentation architectures [37], and later by integrating a semantic segmentation branch with a feature pyramid network (FPN) into Mask R-CNN [36]. Other works [13, 78] extend this idea by relying on specialized architectures and loss functions from both segmentation fields. More recent works [8, 14, 15, 71, 80, 83] handle *things* and *stuff* categories in a unified way via object queries and Hungarian matching [41]. These works generally depend on specialized modules – such as anchor boxes [57], non-maximum suppression [6, 31], merging heuristics [37, 78], or bipartite matching algorithms [8, 14, 71], *etc.* – to generate panoptic masks. Instead, we propose a task-agnostic generative framework to bypass these components. Consequently, we refrain from using task-specific augmentations, such as large-scale jittering or copy-paste augmentations [23].

**General-purpose Frameworks.** Similar to our work, a few task-agnostic solutions have been suggested that cast vision problems as a generative process. Kolesnikov *et al.* [40] minimized task-specific knowledge by learning task-specific guiding codes with an LLM, to train a separate vision model. Chen *et al.* [11] simplified this procedure by framing vision tasks as language modeling tasks, within a single model. Lu *et al.* [47] also follow this route and show promising results for a large variety of vision and language tasks using a unified framework. Each work presents the input as a sequence of discrete tokens which are subsequently reconstructed via autoregressive modeling. Other works [4, 50, 72] leverage masked image modeling to train a single model for multiple vision tasks. Differently, we leverage the denoising process in continuous latent space, which is well-suited to handle dense prediction tasks with high-dimensional inputs [9, 29, 58].

**Denoising Diffusion Models.** Denoising diffusion models [29, 61, 62] were introduced as a new class of generative models. Recent strategies [19, 30, 51, 55] additionally leverage text as guidance, *e.g.*, via CLIP embeddings, to achieve results with impressive realism and control. Building upon its success, a few diffusion-based solutions appeared in the segmentation literature. However, they have undesirable properties: (*i*) the inability to differentiate between instances [2, 3, 5, 32, 76], (*ii*) the necessity for specialized architectures and loss functions [25, 79], or (*iii*) the dependence on object detection weights and bit diffusion [9, 12]. The closest related work from Chen *et al.* [9], presents a framework for panoptic segmentation by leveraging the diffusion process in pixel-space. In contrast, we rigorously follow latent diffusion models by relying on continuous latent codes without the necessity for object detection data. Importantly, we can effortlessly leverage public image-diffusion weights [58] as we keep the architecture task-agnostic.
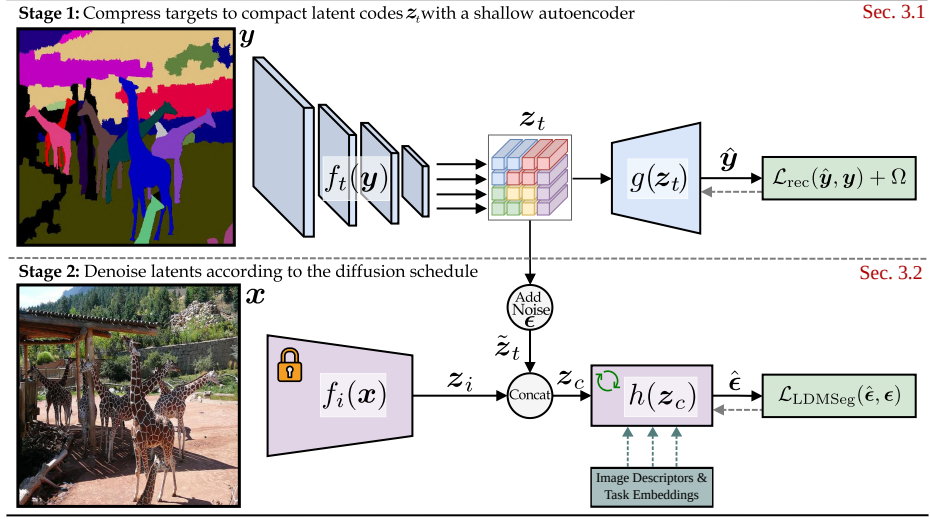
**Fig. 2: Overview of LDMSeg.** Inspired by latent diffusion models, we present a simple diffusion framework for segmentation and mask inpainting. The approach consists of two stages: $(i)$ learn continuous codes $z_t$ with a shallow autoencoder on the labels (Sec. 3.1); $(ii)$ learn a denoising function conditioned on image latents $z_i$ (Sec. 3.2). In the second stage, the error between the predicted noise $\hat{\epsilon}$ and the applied Gaussian noise $\epsilon$ is minimized. During inference, we traverse the denoising process by starting from Gaussian noise. The models $f_t$ and $f_i$ respectively encode the labels and images. While we rely on the image encoder $f_i$ from Stable Diffusion [58], we focus on $f_t$ and $g$ for segmentation. We aim to prioritize generality by limiting task-specific components.

## 3    Method

**Preliminaries.** This paper aims to train a fully generative model for panoptic segmentation via the denoising diffusion paradigm [29]. While this generative approach results in longer sampling times than discriminative methods, we justify this decision through four different lenses: $(i)$ ease of use – diffusion models omit specialized modules, such as non-maximum suppression or region proposal networks, are stable to train, and exhibit faster sampling than autoregressive models [66]; $(ii)$ compositionality – this generative approach captures complex scene compositions with high realism and diversity, while also enabling image editing [55]; $(iii)$ dataset-agnostic – we rely on spatially structured representations that are not tied to predefined classes or taxonomies; $(iv)$ computational cost – latent diffusion models [58] reduce the computational requirements by modeling latents instead of pixels with an autoencoder [35, 52]. Motivated by these points, we create a framework that builds upon latent diffusion models' generative power [55, 58] for the segmentation task.

**Problem Setup.** The considered segmentation task requires a dataset of images $\mathcal{X} = \{x_1, \ldots, x_n\}$ and corresponding ground truth panoptic segmentation masks

$\mathcal{Y} = \{y_1, \ldots, y_n\}$. We don't make any distinction between *things* or *stuff* classes but handle all classes identically. We start by focusing on predicting panoptic IDs without the classes, rendering the method class-agnostic. Later, we extend our approach to also include class labels via task embeddings.

Assume that all images are of size $H \times W$. We train a segmentation model to realize the mapping $\mathbb{R}^{3 \times H \times W} \rightarrow \mathbb{R}^{N \times H \times W}$. For each pixel the model performs a soft assignment over the instances $\{1, \ldots, N\}$. Let the latent representations $z_i$ and $z_t$ respectively refer to the image and target features after the projection to $D$-dimensional latents $\mathbb{R}^{D \times H/f \times W/f}$, where $f \in \mathbb{N}$ denotes the resizing factor.

From a high-level perspective, our method has two key components: learning the prior over segmentation latents $z_t$ and learning a conditional diffusion process in latent space. First, we train a shallow autoencoder to capture the prior distribution $p(z_t)$ that learns to compress the labels into compact latent codes $z_t$. Second, we train a diffusion process – conditioned on the image and target features $p(y|z_t, z_i)$. This component is responsible for decoding noisy target features, guided by image features $z_i$. We make a similar derivation as [55] but condition the generative process on images. Formally, this two-step procedure allows us to construct the conditional distribution $p(y|x)$ via the chain rule as $p(y|x) = p(y|z_t, z_i) \cdot p(z_t)$. Both terms, $p(z_t)$ and $p(y|z_t, z_i)$, are reflected in our framework as two separate training stages (see Figure 2). In particular, Section 3.1 discusses the learning of the prior via autoencoding and Section 3.2 leverages this prior to train a latent diffusion model.

### 3.1 Stage 1: Compress Targets

In the first step, we train a network to compress the task-specific targets into latent codes. While we will focus on panoptic segmentation masks, we note that a similar strategy is applicable to other dense prediction targets (see Section 4.2).

**Motivation.** The motivation to design a shallow autoencoder stems from the observation that segmentation maps differ fundamentally from images as they are lower in entropy. First, these masks typically contain only a small number of unique values as they only capture the object's general shape and location in the scene. Second, neighboring pixels are strongly correlated and often identical, resulting in largely spatially redundant information. We conclude that the segmentation task only necessitates a shallow network to efficiently compress the task's targets and to reliably capture the prior distribution $p(z_t)$. We hypothesize that this observation holds for a myriad of dense prediction tasks, such as panoptic segmentation, depth prediction, saliency estimation *etc*. This also justifies why we refrain from using more advanced autoencoders that rely on computationally demanding architectures with adversarial or perceptual losses [34, 82], *e.g.*, VQGAN [22], typically used to encode images [58].

**Encoding.** We analyze several encoding strategies to represent the input segmentation map $y$, latents $z_t$ and output $\hat{y}$.
1. Input $y$: Let $N$ denote the maximum number of instances per image. RGB-encoding (3 channels), bit-encoding ($\log_2 N$ channels), one-hot-encoding ($N$

channels), or positional-encoding [70] seem all justifiable options. However, one-hot encoding unnecessarily expands the input dimensions and bit encoding avoids the need for a fixed color palette for semantic segmentation. Hence, we follow [9] in representing the instances as bits.

2. Latent code $z_t$: No vector quantization or dimensionality reduction is applied to the latent codes. We empirically found that directly using the continuous latents resulted in a simple architectural design with a small memory footprint (shallow) while simultaneously demonstrating encouraging segmentation results for both reconstruction and generation.

3. Output $\hat{y}$: The output is one-hot encoded to train the autoencoder with a standard cross-entropy loss. In practice, we define $N$ output channels to reconstruct $N$ different instances in a scene, *e.g.*, 128.

**Architectural Design.** The architecture of the autoencoder follows a simple and shallow design. It comprises an encoder $f_t$ and decoder $g$, yielding the overall function $g \circ f_t$. The encoder $f_t$ includes only a few strided convolutions to compress the targets and is inspired by ControlNet [81]. For instance, targets with size 512×512 can be resized efficiently to 64×64, in order to leverage the latent space of Stable Diffusion [58] by stacking 3 convolutions of stride 2. Similarly, the decoder $g$ consists of one or more transpose convolutions to upscale the masks and minimize a loss at pixel-level. As a consequence, the number of trainable parameters is at least 2 orders of magnitudes smaller than the amount of parameters in the diffusion model $h$ ($\approx$ 2M *vs.* 800M). The model's shallow design brings several advantages to the table: fast training, good generalization across datasets, and applicability to inpainting without architectural changes.

**Loss Function.** The autoencoder aims to minimize the reconstruction error $\mathcal{L}_{\mathrm{rec}}$ between the outputs $\hat{y}$ and the one-hot encoded segmentation masks $y$. While regression losses are a valid choice, we opt for a categorical loss due to the segmentation task's discrete nature. The reconstruction loss comprises two loss terms: ($i$) the cross-entropy loss $\mathcal{L}_{\mathrm{ce}}$ enforces unique and confident predictions for each pixel; ($ii$) the mask loss $\mathcal{L}_{\mathrm{m}}$ further refines the segmentation masks by treating each instance individually, alleviating the need for exhaustively labeled images, in contrast to prior generalists [40, 72]. This term is implemented via the BCE and Dice losses [27, 64]. Note that this autoencoding strategy prevents the need for Hungarian matching [41].

Typically, latent diffusion models incorporate a penalty term $\Omega$ into the loss formulation to align the bottleneck latents with a standard Gaussian distribution $\mathcal{N}(0, I)$. This term generally takes the form of a KL divergence, resulting in a variational autoencoder [35]. However, we empirically observed that weight decay regularization suffices to keep the weights $w$, and by extension the latents $z_t$, bounded (see Section 4.3). Consequently, the magnitude of the model's weights $||w||_2$ is penalized, resulting in the final loss formulation:

$$\begin{aligned}
\mathcal{L}_{\mathrm{AE}}(w; y) &= \mathcal{L}_{\mathrm{rec}}(w; \hat{y}, y) + \Omega(z_t, w) \\
&= \mathcal{L}_{\mathrm{ce}}(w; \hat{y}, y) + \mathcal{L}_{\mathrm{m}}(w; \hat{y}, y) + \lambda ||w||_2^2,
\end{aligned} \tag{1}$$
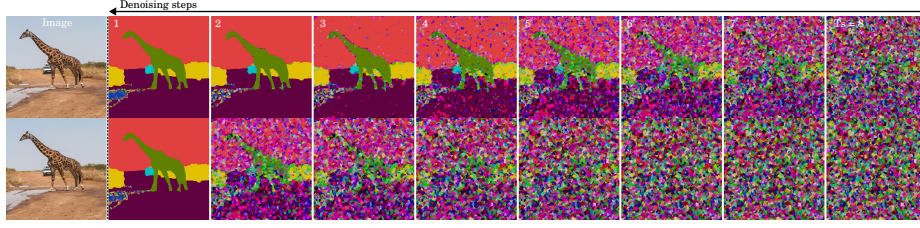
**Fig. 3: Diffusion Process and SNR.** (**1**) **During training** we randomly sample a timestep from $[1, T]$ in the denoising process. We can increase the RGB-image's importance by strengthening the noise: ($i$) Following [9, 58], we downscale the latents $z_c$ using scaling factor $s \in \mathbb{R}$ and demonstrate its impact – Row 1 ($s = 1.0$) is clearly easier to decode than row 2 ($s \approx 0.18$ [58]). ($ii$) Losses for timesteps near 0 are further downscaled to avoid overfitting. Both strategies force the model to focus on the RGB image in generating plausible segmentation maps. Note, we don't apply explicit constraints to the prior distribution $p(z_t)$, *e.g.*, match a standard Gaussian $\mathcal{N}(0, 1)$. (**2**) **During sampling** the denoising process is traversed from right to left in $T_s$ iterations.

where $\hat{y}$ denotes the reconstructed segmentation map. Furthermore, we follow PointRend [39] to select logits that correspond with uncertain regions. This strategy limits the memory consumption as well as the total training time.

### 3.2   Stage 2: Train a Denoising Diffusion Model

**Image-Conditioned Diffusion Process.**  The second stage of the framework models the function $h$ via a conditional diffusion process. We aim to learn a diffusion process over discrete timesteps by conditioning the model on images and noisy segmentation masks for each individual timestep. We further follow the formulation of Stable Diffusion [58]. This process is carried out in a joint latent space – using our trained segmentation encoder $f_t$ and given image encoder $f_i$ [58] – in order to project the targets and images to their respective latent representations $z_t$ and $z_i$. Next, we will discuss the training and inference procedures.

During training we linearly combine the noise $\epsilon$ with the latents $z_t$ for a randomly sampled timestep $j \in [1, T]$:

$$\tilde{z}_t^j = \sqrt{\bar{\alpha}_j} z_t + \sqrt{1 - \bar{\alpha}_j} \epsilon, \tag{2}$$

where $\bar{\alpha}_j$ is defined by the noise schedule, following Rombach *et al.* [58]. The latents are subsequently fused $\{\tilde{z}_t^j, z_i\}$ via channel-wise concatenation as $z_c \in \mathbb{R}^{2D \times H/f \times W/f}$, before feeding them to the UNet [58, 59] $h_\theta$, parameterized with weights $\theta$. Despite the examination of various fusing techniques (*i.e.,* fusing intermediate features via cross-attention or via modality-specific branches) we found that straightforward concatenation at the input works surprisingly well. At the output, the reconstruction error between the added Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ and the predicted noise $\hat{\epsilon}$ is minimized [29] as

$$\mathcal{L}_{\text{LDMSeg}}(\theta; \epsilon) = \mathbb{E}_{z_c, \epsilon \sim \mathcal{N}(0, I), j} \left[ ||\epsilon - h_\theta(z_c, j)||_2^2 \right], \tag{3}$$

---

**Algorithm 1** Forward pass.

```
# f_i, f_t, h: encoders f_i & f_t, UNet h
# x: images of size [bs, 3, H, W]
# y: bit maps of size [bs, log(N), H, W]
# s, T: scaling factor, number of train steps
y = 2 * y - 1,
x = x / 127.5 - 1
z_t = f_t(y) * s
z_i = f_i(x) * s
j = torch.randint((bs,) 0, T)
noise = torch.randn_like(z_t)
# apply Eq. 2
z_t = scheduler.add_noise(z_t, noise, j)
z_c = torch.cat([z_t, z_i], dim=1)
noise_pred = h(z_c, j)
loss = torch.sum((noise_pred - noise)**2, dim
    =[1,2,3])
loss = torch.mean(loss * scheduler.weights[j])
```

scheduler.weights: array of length $T$ with loss weights

**Algorithm 2** Sampling process.

```
# f_i, g: image and segmentation decoders
# h: denoising UNet
# x: image of size [3, H, W]
# T, Ts: #training steps, #inference steps
x = x / 127.5 - 1
z_i = f_i(x) * s
z_t = torch.randn_like(z_i) # Gaussian noise

for j in scheduler.inference_steps:
    z_c = torch.cat([z_t, z_i], dim=0)
    noise_pred = h(z_c, j)
    j_prev = j - T // Ts
    # apply Eq. 4
    z_t = scheduler.step(z_t, noise_pred, j,
        j_prev)
y_pred = g(z_t) # decode latents
```

scheduler.inference_steps: chosen sampling timesteps
scheduler.step: function - predict previous sample

---

where each timestep $j$ is uniformly sampled from $[1, T]$. To reduce training time, we downscale the loss for small timesteps, *i.e.* $j < 25\% \cdot T$. These latents have a high signal-to-noise ratio (SNR) and are thus relatively easy to denoise without modeling semantics. We refer to Algorithm 1 for an overview of the forward pass.

During sampling, the denoising process is traversed from right to left (see Figure 3). It starts from Gaussian noise and progressively adds more details to the segmentation map as controlled by the input image. We can rely on the DDIM scheduler [62] to apply this denoising process over a small number of sampling timesteps $T_s << T$. Here, the previous sample is computed as

$$\tilde{z}_t^{j-1} = \frac{\sqrt{\bar{\alpha}_{j-1}}}{\sqrt{\bar{\alpha}_j}}(\tilde{z}_t^j - \sqrt{1 - \bar{\alpha}_j}) \cdot \hat{\epsilon} + \sqrt{1 - \bar{\alpha}_{j-1}} \cdot \hat{\epsilon}, \tag{4}$$

which follows directly from Equation 2. Recall that this strategy allows us to model $p(y|z_t, z_i)$. Finally, Algorithm 2 provides the details of the sampling process and Figure 3 serves as an illustration.

**Image Descriptors.** Complementary, we briefly experimented with adding guidance via Stable Diffusion's cross-attention layers to improve sample quality. This mechanism was initially intended for text embeddings, but can be used to feed any image descriptor. We didn't immediately observe improvements with self/weakly-supervised priors, like image embeddings from CLIP [54], or when using captions from BLIP [42]. Hence, we can bypass the cross-attention layers with a skip connection in the class-agnostic setup.

**Task Embeddings.** To extend our framework to multiple tasks, we add learnable embeddings. We query the model for a certain task via its cross-attention layers. In particular, (class-aware) panoptic masks can be obtained by merging the semantic and instance predictions, which necessitates two task embeddings.

### 3.3   Segmentation Mask Inpainting

**Setup.** Our image-conditioned diffusion model is well-suited to complete partial segmentation masks. This is potentially useful for completing sparse segmentation

masks obtained from projected point clouds or for interactive image labeling with rough brush strokes. In contrast to image editing [58], the considered segmentation maps contain empty regions, which we initialize with 0's. We simulate different sparsities in Section 4.1. Existing state-of-the-art approaches are not designed for these applications as they decouple the output from the input via bipartite matching [8, 14]. They require additional steps to match predictions with the given partial segmentation IDs (*e.g.*, via majority voting). In contrast, diffusion models act on the corrupted masks by default.

**Inpainting Process.** Ideally, we can tackle inpainting problems out-of-the-box, *i.e.,* without finetuning. To achieve this goal, the inference loop, previously discussed in Algorithm 2, is modified. Assume that we have a dataset of pairs $(y, m)$. Each pair contains a sparse segmentation mask $y \in \{0,1\}^{\log_2 N \times H \times W}$, represented as a bit map, and a valid mask $m \in \{0,1\}^{H \times W}$, represented as a boolean mask. Now, the diffusion process should fill in the missing regions in $y$, determined by the zeros in $m$. At each step of the denoising process, the latents corresponding to the given pixels in $m$ are fixed. The key differences with the sampling process are highlighted in Algorithm 3.

**Algorithm 3** Inpainting process.

```
# f_i, f_t: image encoder, segm. encoder
# g, h: segmentation decoder, denoising UNet
# x: image of size [3, H, W]
# y: sparse bit map of size [log(N), H, W]
# m: boolean mask w/ valid pixels, size [H, W]
# T, Ts: #train steps, #inference steps
y = 2 * y - 1
y[:, ~ m] = 0 # set invalid regions to 0
x = x / 127.5 - 1
z_i = f_i(x) * s
z_t_masked = f_t(y) * s
m = interpolate(m, size=z_i.shape[-2:])
z_t = torch.randn_like(z_i)
for j in scheduler.inference_steps:
    z_c = torch.cat([z_t, z_i], dim=0)
    noise_pred = h(z_c, j)
    # apply Eq. 2 (solve for z_t)
    z_t = scheduler.remove_noise(z_t,
        noise_pred, j)
    z_t[:, m] = z_t_masked[:, m] # keep latents
    j_prev = j - T // Ts
    # apply Eq. 2
    z_t = scheduler.add_noise(z_t, noise_pred,
        j_prev)
y_pred = g(z_t) # decode latents
```

## 4 Experiments

**Dataset.** We conduct the bulk of our experiments on COCO [43] by relying on the panoptic masks with *stuff* and *things* classes. It contains 118k and 5k images for training and evaluation respectively.

**Architecture and Training Setup.** The maximum number of detectable segments $N$ is set to 256. We resize the input to $512 \times 512$, apply random horizontal flipping, and randomly assign integers from $[0, N-1]$ to the segments. The segmentation encoder $f_t$ processes the panoptic mask $y$ by leveraging 3 convolutional layers with stride 2 and SiLU [21] activations. This results in a resizing factor $f$ of 8 and latents with size $4 \times 64 \times 64$. We rely on the image encoder $f_i$ from Rombach *et al.* [58] (VAE) to convert the image $x$ into latents. We adopt Stable Diffusion's pretrained weights to initialize the UNet $h$, and its rescaling factor $s$ to lower the SNR as $s \cdot z_c$ [58]. Notably, 4 zero-initialized channels are appended to the first convolutional layer of $h$, allowing us to operate on the concatenated input $z_c$. Further, self-conditioning [12] is used to improve

**Fig. 4: Qualitative Results.** The figure displays results on COCO `val2017`. We follow the inference setup (Section 3.2) to sample from our model. Only the arg max operator is applied for post-processing. Our model disentangles overlapping instances in challenging scenes without complex modules or post-processing. To visualize, segments are assigned to random colors, and missing (VOID) pixels in the ground truth are black.

sample quality. Our segmentation decoder $g$ consists of 2 transpose convolutions, resulting in an upscaling factor of 4. We randomly sample $j$ from 1000 discrete timesteps and linearly decay the loss for the bottom 25% to lower the impact of samples with high SNR. The AdamW [46] optimizer is adopted with a learning rate of $1e^{-4}$ and weight decay of $1e^{-1}$. Finally, the first stage is trained for 60k iterations with a batch size of 8 while the second stage is trained for 50k iterations with a batch size of 256 unless stated otherwise.

**Inference Setup and Evaluation Protocol.** During inference, the DDIM scheduler [63] generates samples with 50 equidistant timesteps in latent space. At the end of the denoising process, we decode and upscale the segmentation logits (output of $g$) with a factor of 2 using bilinear interpolation. The arg max operator produces the final (discrete) per-pixel segmentation masks. We benchmark our approach with the Panoptic Quality (PQ) evaluation protocol, defined by Kirillov *et al.* [37]. PQ is the product of two quality metrics: the segmentation quality which measures the intersection-over-union of matched segments (IoU) and the recognition quality which measures the precision and recall (F-score).

### 4.1    Segmentation Results

**Image-conditioned Mask Generation.** The model directly divides an image into non-overlapping semantic masks, capturing different objects in the scene, such as persons, horses and cars. Figure 4 showcases the predictions.
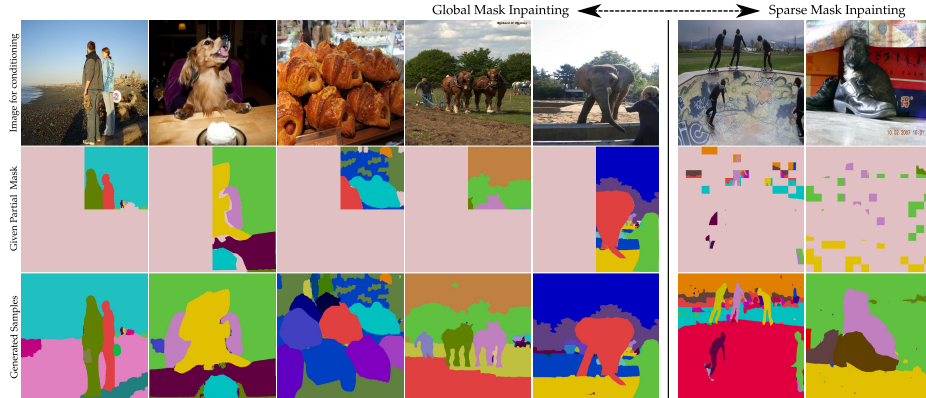
Fig. 5: **Mask Inpainting.** The figure visualizes generated samples for different granularity levels by following Section 3.3. The model can fill in missing regions by propagating the partially given (random) segmentation IDs using an image-conditioned diffusion process. Global mask inpainting (*left*) results are reasonable out-of-the-box while sparse mask inpainting (*right*) shows inaccuracies. We hypothesize that this can be addressed by further finetuning LDMseg on sparse inpainting masks.

As our model is inherently stochastic, we show the predictions when sampling different seeds in Figure 6 (first column). Rows 2 and 3 show the predictions when starting from the same noise map. The key observation is that different maps generate different segmentation IDs. Compared to other frameworks, the sampled noise resem-



Fig. 6: **Impact of Gaussian Noise.**

bles the object queries in Mask2Former or DETR [8,14], and the regions proposals in Mask R-CNN [27]. Table 1 quantitatively compares LDMSeg with prior art for the class-agnostic setup. We obtain 50.8% PQ on COCO `val`. However, there is still a gap with specialized methods, *e.g.*, Mask2Former [14] reaches 59.0% PQ.

**Mask Inpainting.** Next, we mask random regions in the ground truth to simulate *global* and *local* completion tasks. Figure 5 shows the inpainting performance for two different granularity levels in mask $m$. The results demonstrate that LDMSeg is able to complete panoptic masks via the diffusion process, and without requiring additional components. Figure 9 displays the PQ metric for a wide range of drop rates $d$ and block sizes $B$. While our model is able to complete global regions, there is still room for improvement when considering sparse inpainting tasks (*e.g.*, dropping $16 \times 16$ pixels with a drop rate of 90%). For drop rates above 70% and small block sizes, the reduction rate (negative slope) in PQ increases. Further improvements can be obtained by finetuning or modifying the autoencoder. For instance, shallower encoders $f_t$ could adapt better to sparse

**Table 1: Class-agnostic Comparison.** We compare with Mask2Former on COCO `val2017`. Training details for Mask2Former are specified in Supplement A, following SAM [38] with MAE [26] or DINOv2 [53] (⋆) initialization. (†) denotes the adoption of Stable Diffusion's VAE with 84M parameters, resulting in 40× more parameters (84M *vs.* 2M) for the same performance (Section 3.1). $PQ_{inpaint}$ is the averaged PQ metric when dropping $16 \times 16$ pixels with probabilities from 10 to 90%.

| Method | Backbone | PQ ↑ | $PQ_{inpaint}$ ↑ |
|---|---|---|---|
| *Specialist approaches:* | | | |
| MaskFormer [15] | ViT [20] | 54.1 | ✗ |
| Mask2Former [14] | ViT [20] | 56.5 | ✗ |
| Mask2Former⋆ [14] | ViT [20] | 59.0 | ✗ |
| *Generalist approaches:* | | | |
| **LDMSeg**† | UNet [59] | **50.9** | – |
| **LDMSeg** | UNet [59] | 50.8 | **61.3** |

**Table 2: State-of-the-art Comparison.** The table presents the panoptic and semantic segmentation results on COCO `val2017` and ADE20k `val` respectively.

| Method | Backbone | COCO [43]<br>PQ ↑ | ADE20k [84]<br>mIoU ↑ |
|---|---|---|---|
| *Specialist approaches:* | | | |
| PanopticFPN [36] | ResNet [28] | 44.1 | – |
| DETR [8] | ResNet [28] | 45.6 | – |
| MaskFormer [15] | ResNet [28] | 46.5 | 44.5 |
| UPerNet [77] | Swin-L [44] | ✗ | 52.1 |
| DDP [32] | Swin-L [44] | ✗ | 53.2 |
| Mask2Former [14] | Swin-L [44] | **57.8** | **56.1** |
| *Generalist approaches:* | | | |
| Painter [72] | ViT [20] | 41.3 | 47.3 |
| UViM [40] | ViT [20] | 43.1 (45.8) | – |
| **LDMSeg** | UNet [59] | **44.3** | **52.2** |

inputs, in exchange for lower reconstruction quality. Others [14, 40, 72] are not designed to complete partial masks without complex post-processing.

**State-of-the-art Comparison.** As the panoptic task combines instance and semantic segmentation, we introduce two learnable task embeddings. Additionally, we adopt a ViT-B [20] as the image encoder to improve the performance (see Supplement B for a component analysis). Table 2 compares LDMSeg with state-of-the-art methods after training for 100k iterations:
- *LDMSeg* vs. *Specialists*: Our generative approach is competitive with several specialized approaches [8, 36, 78]. For instance, we can match the performance of PanopticFPN [37] while not requiring region proposals. Instead, we rely on the diffusion process to solve the permutation-invariance of the instances.
- *LDMSeg* vs. *Generalists*: (*i*) Painter necessitates two separate encoding schemes for *things* and *stuff* categories, as well as non-maximum-suppression

**Table 3: Multi-Task Performance.** The table reports results for semantic segmentation, panoptic segmentation and relative depth (between 0 - 1) on COCO `val2017`.

| Setting | # Iters | Semantic Seg. mIoU ↑ | Panoptic Seg. PQ ↑ | Relative Depth RMSE ↓ |
|---|---|---|---|---|
| Multi-Task | 50 | 60.1 | 44.1 | 0.075 |
| Single-Task | 50 | 61.5 | 44.3 | 0.067 |

(NMS). LDMSeg outperforms Painter (44.3% *vs.* 41.3%) while not relying on these components, nor on complex merging strategies. This renders our approach more general. In fact, our post-processing time is an order of magnitude smaller (see Supplement B). (*ii*) Additionally, LDMSeg is competitive with UViM's [40] public model (44.3% *vs.* 43.1%). We hypothesize that the disparities in UViM's results could be attributed to its reliance on specific code lengths and code dropout [40], in order to train its autoregressive language model (LM). In contrast, we leverage latent codes centered around zero using a shallow autoencoder, enabling better control. To increase the denoising difficulty, we simply lower the scaling factor (see Section 3.2). Pix2Seq-$\mathcal{D}$ [9] relies on a ResNet backbone that is pretrained with additional bounding-box annotations (Objects365 [60]) and achieves 50.3% PQ. However, its unreleased weights poses a challenge in adopting this paradigm. To our knowledge, LDMSeg is first in demonstrating that a latent diffusion process can bypass object detection data and its related modules. Larger datasets and a higher resolution will help in closing the gap.

Finally, we tackle semantic segmentation and show results for ADE20k [84]. This dataset contains 20k training images and 2k validation images, covering 150 semantic classes. LDMSeg surpasses well-performing methods [15,72,77], reaching 52.2% mIoU on the validation set. Interestingly, our model is able to capture the mapping from latents to the respective classes without making changes to its design. While DDP [32] performs slightly better (52.2% *vs.* 53.2%), it can not handle the permutation-invariance of the instance masks. This limitation stems from relying on the diffusion process as a refinement step and its necessity for a direct mapping between inputs and outputs (*e.g.*, in semantic segmentation).

## 4.2   Multi-Task Learning Results

We now broaden the scope and extend our approach to a multi-task setting. Table 3 reports the results when training on 3 vision tasks for 100k iterations on COCO in total. We handle all tasks identically, including the same scaling factor and augmentations. Given that the single-task setup has been trained for 3 × longer, it serves as an upperbound (*e.g.*, 44.2% *vs.* 44.3% for PQ). Most importantly, we conclude that LDMSeg can be trained on multiple dense prediction tasks simultaneously by leveraging task embeddings. We refer to Supplement A for more details and visualizations.
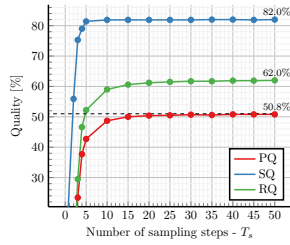
**Fig. 7: Sampling Steps.** The impact of sampling steps on PQ, SQ, and RQ.
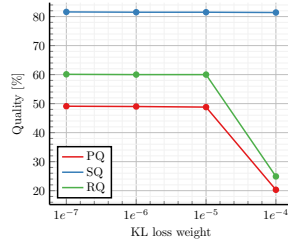
**Fig. 8: KL Loss.** The impact of a KL loss on PQ, SQ, and RQ.
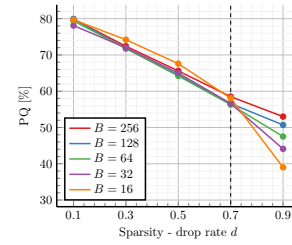
**Fig. 9: Mask Inpainting.** PQ for varying drop rates and block sizes.

### 4.3   Ablation Studies

We ablate LDMSeg in the class-agnostic setting after training for 50k iterations.

– The impact of the number of inference steps is shown in Figure 7. We observe that the PQ starts plateauing at 20 iterations. Longer inference schedules can further improve the recognition quality (a reduction in false positives and negatives), while keeping the segmentation quality constant ($\sim 80\%$).

– Figure 8 demonstrates the impact of adding a KL loss to the autoencoding objective in Equation 1. While it aligns the latents with a standard Gaussian distribution, increasing its loss weight beyond $1e^{-5}$ hurts the reconstruction quality. We conclude that weight decay suffices to keep the latents bounded.

– Table 1 verifies the hypothesis that panoptic masks don't require a powerful VAE. In particular, we finetuned Stable Diffusion's VAE [58] instead of our shallow autoencoder (see Section 3.1). However, this results in a similar PQ (50.9% *vs.* 50.8%). Our shallow autoencoder contains $40\times$ less parameters, which reduces the training time of stage 2 up to 20% (see Section 3.2).

## 5   Conclusion

We presented LDMSeg, a simple yet powerful latent diffusion approach for panoptic segmentation and mask inpainting. In contrast to prior art, we leverage plain latent diffusion models by building upon Stable Diffusion [58]. In summary, the proposed image-conditioned diffusion process has the following advantages: ($i$) it bypasses specialized modules, such as region proposals and bipartite matching; ($ii$) our model unlocks sparse panoptic mask completion without finetuning; ($iii$) the approach can easily be extended to a multi-task setting by introducing task embeddings. The experiments show that LDMSeg is versatile while also outperforming the majority of prior *generalists*. Due to its simple and general design, we believe there is still room for improvement in terms of accuracy and sampling speed. Evident future directions include: training LDMSeg on larger datasets and incorporating more dense prediction tasks (*e.g.*, edge detection). Hence, we hope that this work will spark further interest in designing general-purpose approaches for dense prediction tasks.

# References

1. Abu Alhaija, H., Mustikovela, S.K., Mescheder, L., Geiger, A., Rother, C.: Augmented reality meets computer vision: Efficient data generation for urban driving scenes. International Journal of Computer Vision (IJCV) (2018)
2. Amit, T., Shaharbany, T., Nachmani, E., Wolf, L.: Segdiff: Image segmentation with diffusion probabilistic models. arXiv preprint arXiv:2112.00390 (2021)
3. Asiedu, E.B., Kornblith, S., Chen, T., Parmar, N., Minderer, M., Norouzi, M.: Decoder denoising pretraining for semantic segmentation. arXiv preprint arXiv:2205.11423 (2022)
4. Bar, A., Gandelsman, Y., Darrell, T., Globerson, A., Efros, A.: Visual prompting via image inpainting. In: Advances in Neural Information Processing Systems (NeurIPS) (2022)
5. Baranchuk, D., Rubachev, I., Voynov, A., Khrulkov, V., Babenko, A.: Label-efficient semantic segmentation with diffusion models. In: International Conference on Learning Representations (ICLR) (2022)
6. Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Soft-nms–improving object detection with one line of code. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
7. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
8. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision (ECCV) (2020)
9. Chen, T., Li, L., Saxena, S., Hinton, G., Fleet, D.J.: A generalist framework for panoptic segmentation of images and videos. In: International Conference on Computer Vision (ICCV) (2023)
10. Chen, T., Saxena, S., Li, L., Fleet, D.J., Hinton, G.: Pix2seq: A language modeling framework for object detection. In: International Conference on Learning Representations (ICLR) (2022)
11. Chen, T., Saxena, S., Li, L., Lin, T.Y., Fleet, D.J., Hinton, G.E.: A unified sequence interface for vision tasks. In: Advances in Neural Information Processing Systems (2022)
12. Chen, T., Zhang, R., Hinton, G.: Analog bits: Generating discrete data using diffusion models with self-conditioning. In: International Conference on Learning Representations (ICLR) (2023)
13. Cheng, B., Collins, M.D., Zhu, Y., Liu, T., Huang, T.S., Adam, H., Chen, L.C.: Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
14. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
15. Cheng, B., Schwing, A.G., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. In: Advances in Neural Information Processing Systems (NeurIPS) (2021)
16. Chiu, M.T., Xu, X., Wei, Y., Huang, Z., Schwing, A.G., Brunner, R., Khachatrian, H., Karapetyan, H., Dozier, I., Rose, G., et al.: Agriculture-vision: A large aerial image database for agricultural pattern analysis. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2020)

17. Cho, J.H., Mall, U., Bala, K., Hariharan, B.: Picie: Unsupervised semantic segmentation using invariance and equivariance in clustering. In: CVPR (2021)
18. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
19. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. In: Advances in Neural Information Processing Systems (NeurIPS) (2021)
20. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (ICLR) (2021)
21. Elfwing, S., Uchibe, E., Doya, K.: Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. Neural networks **107**, 3–11 (2018)
22. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
23. Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T.Y., Cubuk, E.D., Le, Q.V., Zoph, B.: Simple copy-paste is a strong data augmentation method for instance segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
24. Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., et al.: Ego4d: Around the world in 3,000 hours of egocentric video. arXiv preprint arXiv:2110.07058 (2021)
25. Gu, Z., Chen, H., Xu, Z., Lan, J., Meng, C., Wang, W.: Diffusioninst: Diffusion model for instance segmentation. arXiv preprint arXiv:2212.02773 (2022)
26. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
27. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: International Conference on Computer Vision (ICCV) (2017)
28. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
29. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems (NeurIPS) (2020)
30. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)
31. Hosang, J., Benenson, R., Schiele, B.: Learning non-maximum suppression. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
32. Ji, Y., Chen, Z., Xie, E., Hong, L., Liu, X., Liu, Z., Lu, T., Li, Z., Luo, P.: Ddp: Diffusion model for dense visual prediction. arXiv preprint arXiv:2303.17559 (2023)
33. Jia, X., De Brabandere, B., Tuytelaars, T., Van Gool, L.: Dynamic filter networks. In: Advances in Neural Information Processing Systems (NeurIPS) (2016)
34. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision (ECCV) (2016)
35. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: International Conference on Learning Representations (ICLR) (2014)
36. Kirillov, A., Girshick, R., He, K., Dollár, P.: Panoptic feature pyramid networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (2019)

37. Kirillov, A., He, K., Girshick, R., Rother, C., Dollár, P.: Panoptic segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
38. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)
39. Kirillov, A., Wu, Y., He, K., Girshick, R.: Pointrend: Image segmentation as rendering. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
40. Kolesnikov, A., Susano Pinto, A., Beyer, L., Zhai, X., Harmsen, J., Houlsby, N.: Uvim: A unified modeling approach for vision with learned guiding codes. In: Advances in Neural Information Processing Systems (NeurIPS) (2022)
41. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955)
42. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: International Conference on Machine Learning (ICML) (2022)
43. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision (ECCV) (2014)
44. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: International Conference on Computer Vision (ICCV) (2021)
45. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
46. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
47. Lu, J., Clark, C., Zellers, R., Mottaghi, R., Kembhavi, A.: Unified-io: A unified model for vision, language, and multi-modal tasks. In: International Conference on Learning Representations (ICLR) (2023)
48. Menze, B.H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R., et al.: The multimodal brain tumor image segmentation benchmark (brats). IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI) (2014)
49. Minaee, S., Boykov, Y.Y., Porikli, F., Plaza, A.J., Kehtarnavaz, N., Terzopoulos, D.: Image segmentation using deep learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI) (2021)
50. Mizrahi, D., Bachmann, R., Kar, O., Yeo, T., Gao, M., Dehghan, A., Zamir, A.: 4m: Massively multimodal masked modeling. In: Advances in Neural Information Processing Systems (NeurIPS) (2023)
51. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In: International Conference on Machine Learning (ICML) (2022)
52. van den Oord, A., Vinyals, O., kavukcuoglu, k.: Neural discrete representation learning. In: Advances in Neural Information Processing Systems (NeurIPS) (2017)
53. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)

54. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning (ICML) (2021)

55. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 (2022)

56. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: International Conference on Machine Learning (ICML) (2021)

57. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (NeurIPS) (2015)

58. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2022)

59. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (2015)

60. Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., Sun, J.: Objects365: A large-scale, high-quality dataset for object detection. In: International Conference on Computer Vision (ICCV) (2019)

61. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International Conference on Machine Learning (ICML) (2015)

62. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: International Conference on Learning Representations (ICLR) (2021)

63. Song, Y., Ermon, S.: Improved techniques for training score-based generative models. In: Advances in Neural Information Processing Systems (NeurIPS) (2020)

64. Sudre, C.H., Li, W., Vercauteren, T., Ourselin, S., Jorge Cardoso, M.: Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support (2017)

65. Tian, Z., Shen, C., Chen, H.: Conditional convolutions for instance segmentation. In: European Conference on Computer Vision (ECCV) (2020)

66. Van Den Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. In: International Conference on Machine Learning (ICML) (2016)

67. Van Gansbeke, W., Vandenhende, S., Georgoulis, S., Van Gool, L.: Revisiting contrastive methods for unsupervised learning of visual representations. In: Advances in Neural Information Processing Systems (NeurIPS) (2021)

68. Van Gansbeke, W., Vandenhende, S., Georgoulis, S., Van Gool, L.: Unsupervised semantic segmentation by contrasting object mask proposals. In: International Conference on Computer Vision (ICCV) (2021)

69. Van Gansbeke, W., Vandenhende, S., Van Gool, L.: Discovering object masks with transformers for unsupervised semantic segmentation. arXiv preprint arXiv:2206.06363 (2022)

70. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems (NeurIPS) (2017)

71. Wang, H., Zhu, Y., Adam, H., Yuille, A., Chen, L.C.: Max-deeplab: End-to-end panoptic segmentation with mask transformers. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
72. Wang, X., Wang, W., Cao, Y., Shen, C., Huang, T.: Images speak in images: A generalist painter for in-context visual learning. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
73. Wang, X., Yu, Z., De Mello, S., Kautz, J., Anandkumar, A., Shen, C., Alvarez, J.M.: Freesolo: Learning to segment objects without annotations. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
74. Wang, X., Zhang, R., Kong, T., Li, L., Shen, C.: Solov2: Dynamic and fast instance segmentation. In: Advances in Neural Information Processing Systems (NeurIPS) (2020)
75. Wang, X., Girdhar, R., Yu, S.X., Misra, I.: Cut and learn for unsupervised object detection and instance segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
76. Wang, Z., Jiang, Y., Lu, Y., He, P., Chen, W., Wang, Z., Zhou, M., et al.: In-context learning unlocked for diffusion models. In: Advances in Neural Information Processing Systems (NeurIPS) (2024)
77. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: European Conference on Computer Vision (ECCV) (2018)
78. Xiong, Y., Liao, R., Zhao, H., Hu, R., Bai, M., Yumer, E., Urtasun, R.: Upsnet: A unified panoptic segmentation network. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
79. Xu, J., Liu, S., Vahdat, A., Byeon, W., Wang, X., De Mello, S.: Open-vocabulary panoptic segmentation with text-to-image diffusion models. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
80. Yu, Q., Wang, H., Qiao, S., Collins, M., Zhu, Y., Adam, H., Yuille, A., Chen, L.C.: k-means mask transformer. In: European Conference on Computer Vision (ECCV) (2022)
81. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: International Conference on Computer Vision (ICCV) (2023)
82. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
83. Zhang, W., Pang, J., Chen, K., Loy, C.C.: K-net: Towards unified image segmentation. In: Advances in Neural Information Processing Systems (NeurIPS) (2021)
84. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. International Journal of Computer Vision (IJCV) (2019)