A Generalist Framework for Panoptic Segmentation of Images and Videos

Ting Chen, Lala Li, Saurabh Saxena, Geoffrey Hinton†, David J. Fleet†
Google Deepmind
{iamtingchen,lala,srbs,geoffhinton,davidfleet}@google.com

Abstract

Panoptic segmentation assigns semantic and instance ID labels to every pixel of an image. As permutations of instance IDs are also valid solutions, the task requires learning of high-dimensional one-to-many mapping. As a result, state-of-the-art approaches use customized architectures and task-specific loss functions. We formulate panoptic segmentation as a discrete data generation problem, without relying on inductive bias of the task. A diffusion model is proposed to model panoptic masks, with a simple architecture and generic loss function. By simply adding past predictions as a conditioning signal, our method is capable of modeling video (in a streaming setting) and thereby learns to track object instances automatically. With extensive experiments, we demonstrate that our simple approach can perform competitively to state-of-the-art specialist methods in similar settings. [1]

1. Introduction

Panoptic segmentation [30] is a fundamental vision task that assigns semantic and instance labels for every pixel of an image. The semantic labels describe the class of each pixel (e.g., sky, vertical), and the instance labels provides a unique ID for each instance in the image (to distinguish different instances of the same class). The task is a combination of semantic segmentation and instance segmentation, providing rich semantic information about the scene.

While the class categories of semantic labels are fixed a priori, the instance IDs assigned to objects in an image can be permuted without affecting the instances identified. For example, swapping instance IDs of two cars would not affect the outcome. Thus, a neural network trained to predict instance IDs should be able to learn a one-to-many mapping, from a single image to multiple instance ID assignments. The learning of one-to-many mappings is challenging and traditional approaches usually leverage a pipeline of multiple stages involving object detection, segmentation,

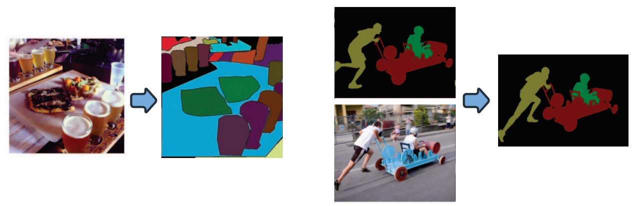


Figure 1: We formulate panoptic segmentation as a conditional discrete mask ($\boldsymbol{m}$) generation problem for images (left) and videos (right), using a Bit Diffusion generative model [12].

merging multiple predictions [30, 34, 66, 40, 14]. Recently, end-to-end methods [58, 17, 70, 16, 35, 68, 69, 33] have been proposed, based on a differentiable bipartite graph matching [7]; this effectively converts a one-to-many mapping into a one-to-one mapping based on the identified matching. However, such methods still require customized architectures and sophisticated loss functions with built-in inductive bias for the panoptic segmentation task.

Eshewing task-specific architectures and loss functions, recent generalist vision models, such as Pix2Seq [10, 11], OFA [60], UViM [31], and Unified I/O [43], advocate a generic, task-agnostic framework, generalizing across multiple tasks while being much simpler than previous models. For instance, Pix2Seq [10, 11] formulates a set of core vision tasks in terms of the generation of semantically meaningful sequences conditioned on an image, and they train a single autoregressive model based on Transformers [55].

Following the same philosophy, we formulate the panoptic segmentation task as a conditional discrete data generation problem, depicted in Figure 1. We learn a generative model for panoptic masks, treated as an array of discrete tokens, conditioned on an input image. One can also apply the model to video data (in an online/streaming setting), by simply including predictions from past frames as an additional conditioning signal. In doing so, the model can then learn to track and segment objects automatically.

Generative modeling for panoptic segmentation is very challenging as the panoptic masks are discrete/categorical

† Equal advising.
[1] Code at https://github.com/google-research/pix2seq

and can be very large. To generate a 512×1024 panoptic mask, for example, the model has to produce more than 1M discrete tokens (of semantic and instance labels). This is expensive for auto-regressive models as they are inherently sequential, scaling poorly with the size of data input. Diffusion models [50, 23, 51, 52] are better at handling high dimension data but they are most commonly applied to continuous rather than discrete domains. By representing discrete data with analog bits [12] we show that one can train a diffusion model on large panoptic masks directly, without the need to also learn an intermediate latent space.

In what follows, we introduce our diffusion-based model for panoptic segmentation, and then describe extensive experiments on both image and video datasets. In doing so we demonstrate that the proposed method performs competitively to state-of-the-art methods in similar settings, proving a simple and generic approach to panoptic segmentation.

2. Preliminaries

Problem Formulation. Introduced in [30], panoptic segmentation masks can be expressed with two channels, $\boldsymbol{m} \in \mathbb{Z}^{H\times W\times 2}$. The first represents the category/class label. The second is the instance ID. Since instance IDs can be permuted without changing the underlying instances, we randomly assign integers in $[0, K]$ to instances every time an image is sampled during training. K is maximum number of instances allowed in any image (0 denotes the null label).

To solve the panoptic segmentation problem, we simply learn an image-conditional panoptic mask generation model by maximizing $\sum_i \log P(\boldsymbol{m}_i|\boldsymbol{x}_i)$, where $\boldsymbol{m}_i$ is a random categorical variable corresponding to the panoptic mask for image $\boldsymbol{x}_i$ in the training data. As mentioned above, panoptic masks may comprise hundreds of thousands or even millions of discrete tokens, making generative modeling very challenging, particularly for autoregressive models.

Diffusion Models with Analog Bits. Unlike autoregressive generative models, diffusion models are more effective with high dimension data [50, 23, 51, 52] . Training entails learning a denoising network. During inference, the network generates target data in parallel, using far fewer iterations than the number of pixels.

In a nutshell, diffusion models learn a series of state transitions to transform noise $\boldsymbol{\epsilon}$ from a known noise distribution into a data sample $\boldsymbol{x}_0$ from the data distribution $p(\boldsymbol{x})$. To learn this mapping, we first define a forward transition from data $\boldsymbol{x}_0$ to a noisy sample $\boldsymbol{x}_t$ as follows,

$$\boldsymbol{x}_t = \sqrt{\gamma(t)}\,\boldsymbol{x}_0 + \sqrt{1-\gamma(t)}\,\boldsymbol{\epsilon}, \tag{1}$$

where $\boldsymbol{\epsilon}$ is drawn from standard normal density, t is from uniform density on [0,1], and $\gamma(t)$ is a monotonically decreasing function from 1 to 0. During training, one learns a neural network $f(\boldsymbol{x}_t, t)$ to predict $\boldsymbol{x}_0$ (or $\boldsymbol{\epsilon}$) from $\boldsymbol{x}_t$, usually formulated as a denoising task with an ℓ_2 loss:

$$\mathcal{L}_{\boldsymbol{x}_0} = \mathbb{E}_{t\sim\mathcal{U}(0,T),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{1}),\boldsymbol{x}_t}\|f(\boldsymbol{x}_t,t)-\boldsymbol{x}_0\|^2. \tag{2}$$

To generate samples from a learned model, it starts with a sample of noise, $\boldsymbol{x}_T$, and then follows a series of (reverse) state transitions $\boldsymbol{x}_T \rightarrow \boldsymbol{x}_{T-\Delta} \rightarrow \cdots \rightarrow \boldsymbol{x}_0$ by iteratively applying the denoising function f with appropriate transition rules (such as those from DDPM [23] or DDIM [51]).

Conventional diffusion models assume continuous data and Gaussian noise, and are not directly applicable to discrete data. To model discrete data, Bit Diffusion [12] first converts integers representing discrete tokens into bit strings, the bits of which are then cast as real numbers (a.k.a., analog bits) to which continuous diffusion models can be applied. To draw samples, Bit Diffusion uses a conventional sampler from continuous diffusion, after which a final quantization step (simple thresholding) is used to obtain the categorical variables from the generated analog bits.

3. Method

We formulate panoptic segmentation as a discrete data generation problem conditioned on pixels, similar to Pix2Seq [10, 11] but for dense prediction; hence we coin our approach Pix2Seq-$\mathcal{D}$. In what follows we first specify the architecture design, and then the training and inference algorithms based on Bit Diffusion.

3.1. Architecture

Diffusion model sampling is iterative, and hence one must run the forward pass of the network many times during inference. Therefore, as shown in Fig. 2, we intentionally separate the network into two components: 1) an image encoder; and 2) a mask decoder. The former maps raw pixel data into high-level representation vectors, and then the mask decoder iteratively reads out the panoptic mask.

Pixel/image Encoder. The encoder is a network that maps a raw image $\boldsymbol{x} \in \mathbb{R}^{H\times W\times 3}$ into a feature map in $\mathbb{R}^{H'\times W'\times d}$ where H' and W' are the height and width of the panoptic mask. The panoptic masks can be the same size or smaller than the original image. In this work, we follow [7, 10] in using ResNet [22] followed by transformer encoder layers [55] as the feature extractor. To make sure the output feature map has sufficient resolution, and includes features at different scales, inspired by U-Net [23, 45, 47] and feature pyramid network [38], we use convolutions with bilateral connections and upsampling operations to merge features from different resolutions. More sophisticated encoders are possible, to leverage recent advances in architecture designs [20, 41, 25, 71, 26], but this is not our main focus so we opt for simplicity.

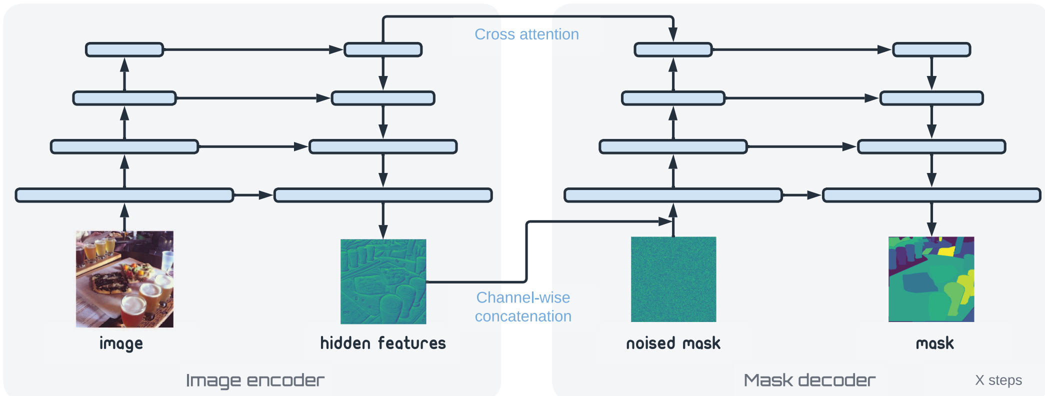


Figure 2: The architecture for our panoptic mask generation framework. We separate the model into image encoder and mask decoder so that the iterative inference at test time only involves multiple passes over the decoder.

Algorithm 1 Pix2Seq-$\mathcal{D}$ training algorithm.

```
def train_loss(images, masks):
  """images: [b, h, w, 3], masks: [b, h', w', 2]."""

  # Encode image features.
  h = pixel_encoder(images)

  # Discrete masks to analog bits.
  m_bits = int2bit(masks).astype(float)
  m_bits = (m_bits * 2 - 1) * scale

  # Corrupt analog bits.
  t = uniform(0, 1)              # scalar.
  eps = normal(mean=0, std=1) # same shape as m_bits.
  m_crpt = sqrt(    gamma(t)) * m_bits +
           sqrt(1 - gamma(t)) * eps

  # Predict and compute loss.
  m_logits, _ = mask_decoder(m_crpt, h, t)
  loss = cross_entropy(m_logits, masks)

  return loss.mean()
```

Algorithm 2 Pix2Seq-$\mathcal{D}$ inference algorithm.

```
def infer(images, steps=10, td=1.0):
  """images: [b, h, w, 3]."""

  # Encode image features.
  h = pixel_encoder(images)

  m_t = normal(mean=0, std=1) # same shape as m_bits.
  for step in range(steps):
    # Get time for current and next states.
    t_now = 1 - step / steps
    t_next = max(1 - (step + 1 + td) / steps, 0)

    # Predict analog bits m_0 from m_t.
    _, m_pred = mask_decoder(m_t, h, t_now)

    # Estimate m at t_next.
    m_t = ddim_step(m_t, m_pred, t_now, t_next)

  # Analog bits to masks.
  masks = bit2int(m_pred > 0)
  return masks
```

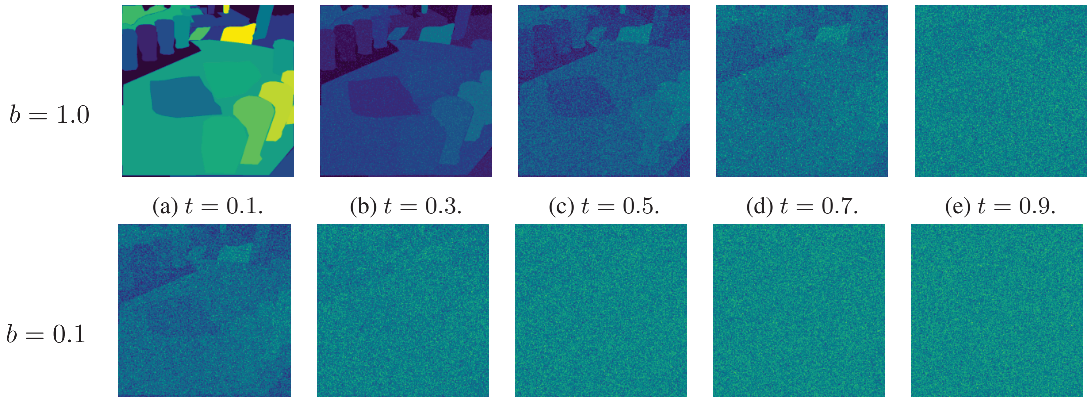
Mask Decoder. The decoder iteratively refines the panoptic mask during inference, conditioned on the image features. Specifically, the mask decoder is a TransUNet [8]. It takes as input the concatenation of image feature map from encoder and a noisy mask (randomly initialized or from previous step), and outputs a refined prediction of the mask. One difference between our decoder and the standard U-Net architecture used for image generation and image-to-image translation [23, 45, 48] is that we use transformer decoder layers on the top of U-Net, with cross-attention layers to incorporate the encoded image features (before upsampling).

3.2. Training Algorithm

Our main training objective is the conditional denoising of analog bits [12] that represent noisy panoptic masks. Algorithm 1 gives the training algorithm (with extra details in A), the key elements of which are introduced below.

Analog Bits with Input Scaling. The analog bits are real numbers converted from the integers of panoptic masks. When constructing the analog bits, we can shift and scale them into $\{-b, b\}$. Typically, b is set to be 1 [12] but we find that adjusting this scaling factor has an significant effect on the performance of the model. This scaling factor effectively allows one to adjust the signal-to-noise ratio of the diffusion process (or the noise schedule), as visualized in Fig. 3. With $b = 1$, we see that even with a large time step $t = 0.7$ (with $t \in [0, 1]$), the signal-to-noise ratio is still relatively high, so the masks are visible to naked eye and the model can easily recover the mask without using the encoded image features. With $b = 0.1$, however, the denoising task becomes significantly harder as the signal-to-noise ratio is reduced. In our study, we find $b = 0.1$ works substantially better than the default of 1.0.

Softmax Cross Entropy Loss. Conventional diffusion models (with or without analog bits) are trained with an ℓ_2 denoising loss. This works reasonably well for panoptic segmentation, but we also discovered that a loss based on softmax cross entropy yields better performance. Al-



(a) $t = 0.1$. (b) $t = 0.3$. (c) $t = 0.5$. (d) $t = 0.7$. (e) $t = 0.9$.

(f) $t = 0.1$. (g) $t = 0.3$. (h) $t = 0.5$. (i) $t = 0.7$. (j) $t = 0.9$.

Figure 3: Noisy masks at different time steps under two input scaling factors, $b = 1.0$ (top row) and $b = 0.1$ (bottom row). Decreasing the input scaling factor leads to smaller signal-to-noise ratio (at the same time step), which gives higher weights to harder cases.

(a) $p = 0.1$. (b) $p = 0.2$. (c) $p = 0.3$. (d) $p = 0.4$.

Figure 4: The effect of p on loss weighting for panoptic masks. With $p = 0$, every mask token is weighted equally (equivalent to no weighting). As p increases, larger weight is given to mask tokens of smaller instances (indicated by warmer colors).

though the analog bits are real numbers, they can be seen as a one-hot weighted average of base categories. For example, '01' $= \alpha_0$'00' $+ \alpha_1$'01' $+ \alpha_2$'10' $+ \alpha_3$'11' where $\alpha_1 = 1$, and $\alpha_0 = \alpha_2 = \alpha_3 = 0$. Instead of modeling the analog bits in '01' as real numbers, with a cross entropy loss, the network can directly model the underlying distribution over the four base categories, and use the weighted average to obtain the analog bits. As such, the mask decoder output not only analog bits (`m_pred`), but also the corresponding logits (`m_logits`), $\tilde{\boldsymbol{y}} \in \mathbb{R}^{H\times W\times K}$, with a cross entropy loss for training; i.e.,

$$\mathcal{L} = \sum_{i,j,k} \boldsymbol{y}_{ijk} \log \text{softmax}(\tilde{\boldsymbol{y}}_{ijk})$$

Here, $\boldsymbol{y}$ is the one-hot vector corresponding to class or instance label. During inference, we still use analog bits from the mask decoder instead of underlying logits for the reverse diffusion process.

Loss Weighting. Standard generative models for discrete data assign equal weight to all tokens. For panoptic segmentation, with a loss defined over pixels, this means that large objects will have more influence than small objects. And this makes learning to segment small instances inefficient. To mitigate this, we use an adaptive loss to improve the segmentation of small instances by giving higher weights to mask tokens associated with small objects. The specific per-token loss weighting is as follows:

$$w_{ij} = 1/c_{ij}^p \text{ , and } w'_{ij} = H * W * w_{ij} / \sum_{ij} w_{ij} \text{ ,}$$

where c_{ij} is the pixel count for the instance at pixel location (i, j), and p is a tunable parameter; uniform weighting occurs when $p = 0$, and for $p > 0$, a higher weight is assigned to mask tokens of small instances. Fig. 4 demonstrate the effects of p in weighting different mask tokens.

3.3. Inference Algorithm

Algorithm 2 summarizes the inference procedure. Given an input image, the model starts with random noise as the initial analog bits, and gradually refines its estimates to be closer to that of good panoptic masks. Like Bit Diffusion [12], we use asymmetric time intervals (controlled by a single parameter `td`) that is adjustable during inference time. It is worth noting that the encoder is only run once, so the cost of multiple iterations depends on the decoder alone.

3.4. Extension to Videos

Our image-conditional panoptic mask modeling with $p(\boldsymbol{m}|\boldsymbol{x})$ is directly applicable for video panoptic segmen-

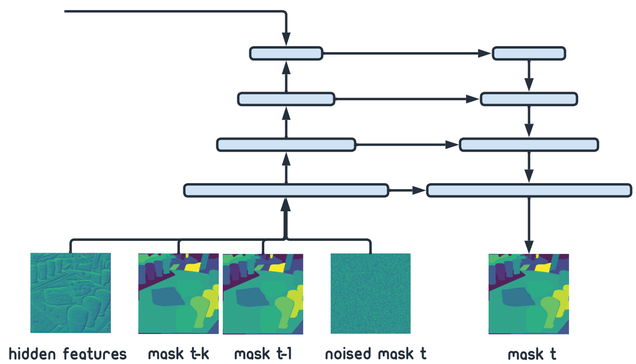


Figure 5: Mask decoder extended for video settings. The image conditional signal to the mask decoder is concatenated with mask predictions from previous frames of the video.

tation by considering 3D masks (with an extra time dimension) given a video. To adapt for online/streaming video settings, we can instead model $p(\boldsymbol{m}_t|\boldsymbol{x}_t, \boldsymbol{m}_{t-1}, \boldsymbol{m}_{t-k})$, thereby generating panoptic masks conditioned on the image and past mask predictions. This change can be easily implemented by concatenating the past panoptic masks $(\boldsymbol{m}_{t-1}, \boldsymbol{m}_{t-k})$ with existing noisy masks, as demonstrated in Fig. 5. Other than this minor change, the model remains the same as that above, which is simple and allows one to fine-tune an image panoptic model for video.

With the past-conditional generation (using the denoising objective), the model automatically learns to track and segment instances across frames, without requiring explicit instance matching through time. Having an iterative refinement procedure also makes our framework convenient to adapt in a streaming video setting where there are strong dependencies across adjacent frames. We expect fewer inference steps to arrive at good segmentation results when there are relative small change in video frames, it may be possible to set refinement steps adaptively across video frames.

4. Experiments

4.1. Setup and Implementation Details

Datasets. For image panoptic segmentation, we conduct experiments on the two commonly used benchmarks: MS-COCO [39] and Cityscapes [19]. MS-COCO contains approximately 118K training images and 5K validation images used for evaluation. Cityscapes contains 2975 images for training, 500 for validation and 1525 for testing. We report results on Cityscapes *val* set, following most existing papers. For expedience we conduct most model ablations on MS-COCO. Finally, for video segmentation we use DAVIS [46] in the challenging unsupervised setting, with no segmentation provided at test time. DAVIS comprises 60 training videos and 30 validation videos for evaluation.

Training. MS-COCO is larger and more diverse than Cityscapes and DAVIS. Thus we mainly train on MS-COCO, and then transfer trained models to Cityscapes and DAVIS with fine-tuning (at a single resolution). We first separately pre-train the image encoder and mask decoder before training the image-conditional mask generation on MS-COCO. The image encoder is taken from the Pix2Seq [10] object detection checkpoint, pre-trained on objects365 [49]. It comprises a ResNet-50 [22] backbone, and 6-layer 512-dim Transformer [55] encoder layers. We also augment image encoder with a few convolutional up-sampling layers to increase its resolution and incorporate features at different layers. The mask decoder is a TransUNet [8] with base dimension 128, and channel multipliers of $1\times$,$1\times$,$2\times$,$2\times$, followed by 6-layer 512-dim Transformer [55] decoder layers. It is pre-trained on MS-COCO as an unconditional mask generation model without images.

Directly training our model on high resolution images and panoptic masks can be expensive as the existing architecture scales quadratically with resolution. So on MS-COCO, we train the model with increasing resolutions, similar to [53, 54, 24]. We first train at a lower resolution (256$\times$256 for images; 128$\times$128 for masks) for 800 epochs with a batch size of 512 and scale jittering [21, 65] of strength $[1.0, 3.0]$. We then continue train the model at full resolution (1024$\times$1024 for images; 512$\times$512 for masks) for only 15 epochs with a batch size of 16 without augmentation. This works well, as both convolution networks and transformers with sin-cos positional encoding generalize well across resolutions. More details on hyper-parameter settings for training can be found in Appendix B.

Inference. We use DDIM updating rules [51] for sampling. By default we use 20 sampling steps for MS-COCO. We find that setting $\mathtt{td}=2.0$ yields near optimal results. We discard instance predictions with fewer than 80 pixels. For inference on DAVIS, we use 32 sampling steps for the first frame and 8 steps for subsequent frames. We set $\mathtt{td}=1$ and discard instance predictions with fewer than 10 pixels.

4.2. Main Results

We compare with two families of state-of-the-art methods, i.e., specialist approaches and generalist approaches. Table 1 summarizes results for MS-COCO. Pix2Seq-$\mathcal{D}$ achieves competitive Panoptic Quality (PQ) to state-of-the-art methods with the ResNet-50 backbone. When compared with other recent generalist models such as UViM [31], our model performs significantly better while being much more efficient. Similar results are obtained for Cityscape, the details of which are given in Appendix C.

Table 2 compares Pix2Seq-$\mathcal{D}$ to state-of-the-art methods on unsupervised video object segmentation on DAVIS, using the standard $\mathcal{J}\&\mathcal{F}$ metrics [46]. Baselines do not in-

Method	Backbone	# of Params	PQ	PQ^{thing}	PQ^{stuff}
Specialist approaches:					
MaskFormer [17]	ResNet-50	45M	46.5	51.0	39.8
K-Net [70]	ResNet-50	-	47.1	51.7	40.3
CMT-DeepLab [68]	ResNet-50	-	48.5	-	-
Panoptic SegFormer [35]	ResNet-50	51M	49.6	54.4	42.4
Mask2Former [16]	ResNet-50	44M	51.9	57.7	43.0
kMaX-DeepLab [69]	ResNet-50	57M	53.0	58.3	44.9
DETR [7]	ResNet-101	61.8M	45.1	50.5	37.0
Mask2Former [13]	Swin-L	216M	57.8	-	-
kMaX-DeepLab [69]	ConvNeXt-L	232M	58.1	64.3	48.8
MasK DINO [33]	Swin-L	223M	59.4	-	-
Generalist approaches:					
UViM [31]	ViT	939M	45.8	-	-
Pix2Seq-$\mathcal{D}$ (steps=5)	ResNet-50	94.5M	47.5	52.2	40.3
Pix2Seq-$\mathcal{D}$ (steps=10)	ResNet-50	94.5M	49.4	54.4	41.9
Pix2Seq-$\mathcal{D}$ (steps=20)	ResNet-50	94.5M	50.3	55.3	42.9
Pix2Seq-$\mathcal{D}$ (steps=50)	ResNet-50	94.5M	50.2	55.1	42.8

Table 1: Results on MS-COCO. Pix2Seq-$\mathcal{D}$ achieves competitive results to state-of-the-art specialist models with ResNet-50 backbone.

Method	Backbone	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$-Mean	$\mathcal{J}$-Recall	$\mathcal{F}$-mean	$\mathcal{F}$-Recall
Specialist approaches:						
RVOS [56]	ResNet-101	41.2	36.8	40.2	45.7	46.4
STEm-Seg [3]	ResNet-101	64.7	61.5	70.4	67.8	75.5
MAST [32]	ResNet-18	65.5	63.3	73.2	67.6	77.7
UnOVOST [44]	ResNet-101	67.9	66.4	76.4	69.3	76.9
Propose-Reduce [37]	ResNeXt-101	70.4	67.0	-	73.8	-
Generalist approaches:						
Pix2Seq-$\mathcal{D}$ (ours)	ResNet-50	68.4	65.1	70.6	71.7	77.1

Table 2: Results of unsupervised video object segmentation on DAVIS 2017 validation set.

clude other generalist models as they are not directly applicable to the task. Our method achieves results on par with state-of-the-art methods without specialized designs.

4.3. Ablations on Training

Ablations on model training are performed with MS-COCO dataset. To reduce the computation cost while still be able to demonstrate the performance differences in design choices, we train the model for 100 epochs with a batch size of 128 in a single-resolution stage (512×512 image size, and 256×256 mask size).

Input Scaling of Analog Bits. Table 3 shows the dependence of PQ results on input scaling of analog bits. The scale factor of 0.1 used here yield results that outperform the standard scaling of 1.0 in previous work [12].

Loss Functions. Table 4 compares our proposed cross entropy loss to the ℓ_2 loss normally used by diffusion models. Interestingly, the cross entropy loss yields substantial gains over ℓ_2.

Loss weighting. Table 5 shows the effects of p for loss weighting. Weighting with $p = 0.2$ appears near optimal and clearly outperforms uniform weighting ($p = 0$).

Input scaling	0.03	0.1	0.3	1.0
PQ	40.8	43.9	38.7	21.3

Table 3: Ablation on input scaling

Loss function	ℓ_2 Regression	Cross Entropy
PQ	41.9	43.9

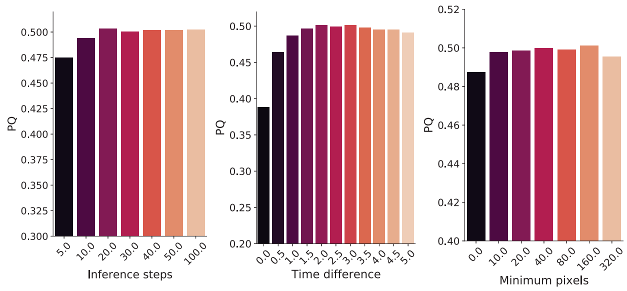
Table 4: Ablation on loss function.

Loss weight p	0	0.2	0.4	0.6
PQ	40.4	43.9	43.7	41.3

Table 5: Ablation on loss weighting.

4.4. Ablations on Inference

Figure 6 explores the dependence of model performance on hyper-parameter choices during inference (sampling),



(a) (b) (c)

Figure 6: Inference ablations on MS-COCO.

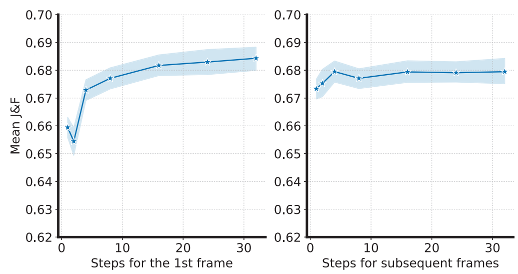


Figure 7: Effect of inference steps on DAVIS. Left: we vary the number of steps for the 1st frame while using a fixed of 8 steps for the remaining frames. Right: we use 8 steps for the 1st frame, while varying the number of steps for the remaining frames. The first frame requires more inference steps due to the cold start.

namely, the number of inference steps, time differences and threshold on the minimum size of instance regions, all on MS-COCO. Specifically, Fig. 6a shows that inference steps of 20 seems sufficient for near optimal performance on MS-COCO. Fig. 6b shows that the `td` parameter as in asymmetric time intervals [12] has a significant impact, with intermediate values (e.g., 2-3) yielding the best results. Fig. 6c shows that the right choice of threshold on small instances leads to small performance gains.

Figure 7 shows how performance varies with the number of inference steps for the first frame, and for the remaining frames, on DAVIS video dataset. We find that more inference steps are helpful for the first frame compared to subsequent frames of video. Therefore, we can reduce the total number of steps by using more steps for the first frames (e.g., 32), and fewer steps for subsequent frames (e.g., 8). It is also worth noting that even using 8 steps for the first frame and only 1 step for each subsequent frame, the model still achieves an impressive $\mathcal{J}\&\mathcal{F}$ of 67.3.

4.5. Case study

Figure 8, 9 and 10 show example results of Pix2Seq-$\mathcal{D}$ on MS-COCO, Cityscape, and DAVIS. One can see that our model is capable of capturing small objects in dense scene well. More visuslizations are shown in Appnedix E.

5. Related Work

Image Panoptic Segmentation. Panoptic segmentation, introduced in [30], unifies semantic segmentation and instance segmentation. Previous approaches to panoptic segmentation involve pipelines with multiple stages, such as object detection, semantic and instance segmentation, and the fusion of separate predictions [30, 34, 66, 40, 14, 7]. With multiple stages involved, learning is often not end-to-end. Recent work has proposed end-to-end approaches with Transformer based architectures [58, 17, 70, 16, 35, 68, 69, 33], for which the model directly predicts segmentation masks and optimizes based on a bipartite graph matching loss. Nevertheless, they still require customized architectures (e.g., per instance mask generation, and mask fusion module). Their loss functions are also specialized for optimizing metrics used in object matching.

Our approach is a significant departure to existing methods with task-specific designs, as we simply treat the task as image-conditional discrete mask generation, without reliance on inductive biases of the task. This results in a simpler and more generic design, one which is easily extended to video segmentation with minimal modifications.

Video Segmentation. Among the numerous video segmentation tasks, video object segmentation (VOS) [46, 61] is perhaps most canonical task, which entails the segmentation of key objects (of unknown categories). Video instance segmentation (VIS) [67] is similar to VOS, but requires category prediction of instances. Video panoptic segmentation (VPS) [28] is a direct extension of image panoptic segmentation to the video domain. All video segmentation tasks involve two main challenges, namely, segmentation and object tracking. And like image segmentation, most existing methods are specialist models comprising multiple stages with pipe-lined frameworks, e.g., track-detect [67, 57, 36, 6, 44], clip-match [3, 5], propose-reduce [37]. End-to-end approaches have also been proposed recently [62, 29], but with a specialized loss function.

In this work we directly take the Pix2Seq-$\mathcal{D}$ model, pretrained on COCO for panoptic segmentation, and fine-tune it for unsupervised video object segmentation (UVOS), where it performs VOS without the need for manual initialization. Model architectures, training losses, input augmentations and sampling methods all remained largely unchanged when applied to UVOS data. Because of this, we believe it is just as straightforward to apply Pix2Seq-$\mathcal{D}$ to the other video segmentation tasks as well.

Others. Our work is also related to recent generalist vision models [10, 11, 60, 31, 43] where both architecture and loss functions are task-agnostic. Existing generalist models are based on autoregressive models, while our work is based on Bit Diffusion [12, 23, 50, 51]. Diffusion models have

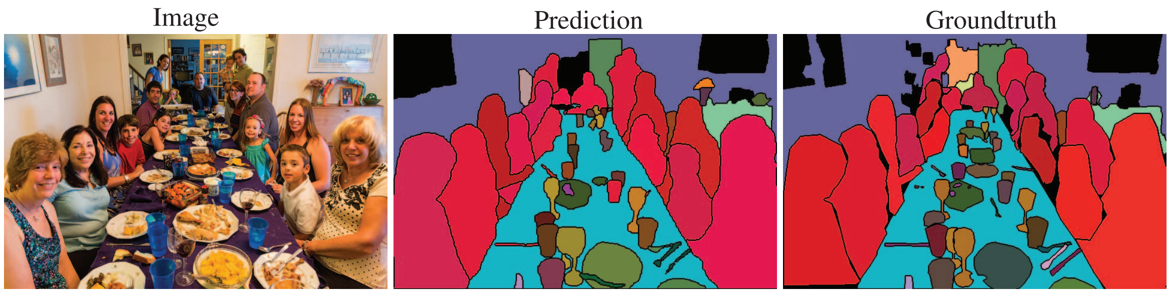


Figure 8: Predictions on MS-COCO *val* set.

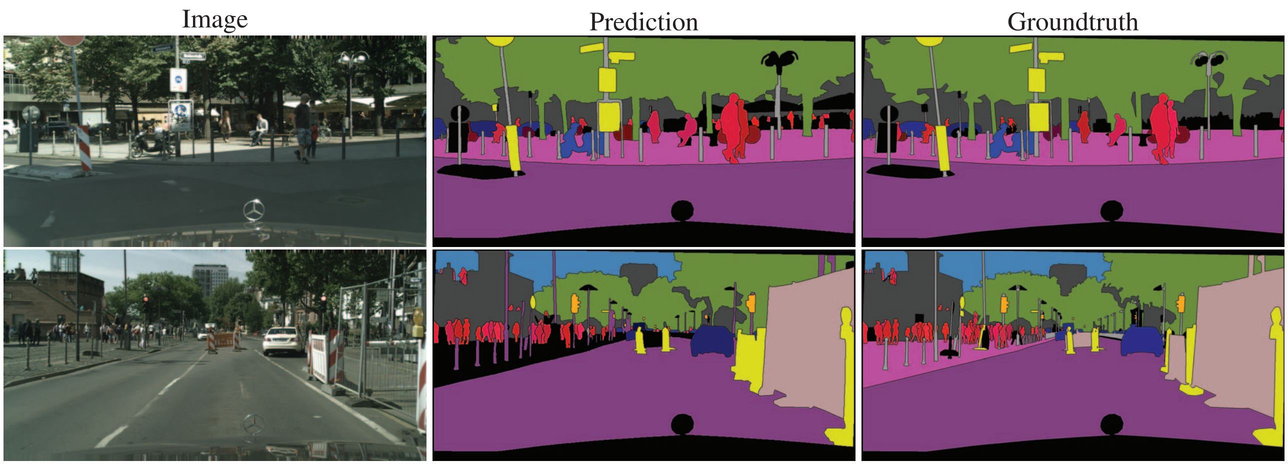


Figure 9: Predictions on Cityscapes *val* set.

Figure 10: Predictions on DAVIS *val* set.

been applied to semantic segmentation, directly [1, 64, 27] or indirectly [4, 2]. However none of these methods model segmentation masks as discrete/categorical tokens, nor are their models capable of video segmentation.

6. Conclusion and Future Work

This paper proposes a simple framework for panoptic segmentation of images and videos, based on conditional generative models of discrete panoptic masks. Our approach is able to model large number of discrete tokens (10^6 in our experiments), which is difficult with other existing generative segmentation models. We believe both the architecture, modeling choices, and training procedure (including augmentations) we use here can be further improved to boost the performance. Furthermore, the required inference steps can also be further reduced with techniques like progressive distillation. Finally, we want to note that, as a significant departure to status quo, we acknowledge that our current empirical result is still behind compared to well-tuned pipelines in existing systems (though our results are still competitive and at a practically usable level). However, with the simplicity of the proposed approach, we hope it would spark future development that drives new state-of-the-art systems.

Acknowledgements

We specially thank Liang-Chieh Chen for the helpful feedback on our initial draft.

References

[1] Tomer Amit, Eliya Nachmani, Tal Shaharbany, and Lior Wolf. Segdiff: Image segmentation with diffusion probabilistic models. *arXiv preprint arXiv:2112.00390*, 2021.

[2] Emmanuel Brempong Asiedu, Simon Kornblith, Ting Chen, Niki Parmar, Matthias Minderer, and Mohammad Norouzi. Decoder denoising pretraining for semantic segmentation. In *Computer Vision and Pattern Recognition workshop*, 2022.

[3] Ali Athar, Sabarinath Mahadevan, Aljosa Osep, Laura Leal-Taixé, and Bastian Leibe. Stem-seg: Spatio-temporal embeddings for instance segmentation in videos. In *European Conference on Computer Vision*, pages 158–177. Springer, 2020.

[4] Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. *arXiv preprint arXiv:2112.03126*, 2021.

[5] Gedas Bertasius and Lorenzo Torresani. Classifying, segmenting, and tracking object instances in video with mask propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9739–9748, 2020.

[6] Jiale Cao, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. Sipmask: Spatial information preservation for fast image and video instance segmentation. In *European Conference on Computer Vision*, pages 1–18. Springer, 2020.

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[8] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L. Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021.

[9] Liang-Chieh Chen, Huiyu Wang, and Siyuan Qiao. Scaling wide residual networks for panoptic segmentation. *arXiv:2011.11675*, 2020.

[10] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. *arXiv preprint arXiv:2109.10852*, 2021.

[11] Ting Chen, Saurabh Saxena, Lala Li, Tsung-Yi Lin, David J Fleet, and Geoffrey Hinton. A unified sequence interface for vision tasks. *arXiv preprint arXiv:2206.07669*, 2022.

[12] Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022.

[13] Bowen Cheng, Anwesa Choudhuri, Ishan Misra, Alexander Kirillov, Rohit Girdhar, and Alexander G Schwing. Mask2former for video instance segmentation. *arXiv preprint arXiv:2112.10764*, 2021.

[14] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12475–12485, 2020.

[15] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation. In *CVPR*, 2020.

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

[17] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021.

[18] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017.

[19] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[21] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2918–2928, 2021.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[24] Jeremy Howard. Training imagenet in 3 hours for 25 minutes. *https://www.fast.ai/2018/04/30/dawnbench-fastai/*, 2018.

[25] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021.

[26] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022.

[27] Boah Kim, Yujin Oh, and Jong Chul Ye. Diffusion adversarial representation learning for self-supervised vessel segmentation. *arXiv preprint arXiv:2209.14566*, 2022.

[28] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Video panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9859–9868, 2020.

[29] Dahun Kim, Jun Xie, Huiyu Wang, Siyuan Qiao, Qihang Yu, Hong-Seok Kim, Hartwig Adam, In So Kweon, and Liang-Chieh Chen. Tubeformer-deeplab: Video mask transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13914–13924, 2022.

[30] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019.

[31] Alexander Kolesnikov, André Susano Pinto, Lucas Beyer, Xiaohua Zhai, Jeremiah Harmsen, and Neil Houlsby. Uvim: A unified modeling approach for vision with learned guiding codes. *arXiv preprint arXiv:2205.10337*, 2022.

[32] Zihang Lai, Erika Lu, and Weidi Xie. Mast: A memory-augmented self-supervised tracker. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[33] Feng Li, Hao Zhang, Shilong Liu, Lei Zhang, Lionel M Ni, Heung-Yeung Shum, et al. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. *arXiv preprint arXiv:2206.02777*, 2022.

[34] Yanwei Li, Xinze Chen, Zheng Zhu, Lingxi Xie, Guan Huang, Dalong Du, and Xingang Wang. Attention-guided unified network for panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7026–7035, 2019.

[35] Zhiqi Li, Wenhai Wang, Enze Xie, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, Ping Luo, and Tong Lu. Panoptic segformer: Delving deeper into panoptic segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1280–1289, 2022.

[36] Chung-Ching Lin, Ying Hung, Rogerio Feris, and Linglin He. Video instance segmentation tracking with a modified vae architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13147–13157, 2020.

[37] Huaijia Lin, Ruizheng Wu, Shu Liu, Jiangbo Lu, and Jiaya Jia. Video instance segmentation with a propose-reduce paradigm. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1739–1748, October 2021.

[38] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[39] 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 *European conference on computer vision*, pages 740–755. Springer, 2014.

[40] Huanyu Liu, Chao Peng, Changqian Yu, Jingbo Wang, Xu Liu, Gang Yu, and Wei Jiang. An end-to-end network for panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6172–6181, 2019.

[41] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[42] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *CVPR*, 2022.

[43] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. *arXiv preprint arXiv:2206.08916v2*, 2022.

[44] Jonathon Luiten, Idil Esen Zulfikar, and Bastian Leibe. Unovost: Unsupervised offline video object segmentation and tracking. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[45] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *arXiv 2102.09672*, 2021.

[46] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016.

[47] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[48] Chitwan Saharia, William Chan, Huiwen Chang, Chris A Lee, Jonathan Ho, Tim Salimans, David J Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. *SIGGRAPH*, 2022.

[49] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019.

[50] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[51] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[52] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[53] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning*, pages 10096–10106. PMLR, 2021.

[54] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy. *Advances in neural information processing systems*, 32, 2019.

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[56] Carles Ventura, Miriam Bellver, Andreu Girbau, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. Rvos: End-to-end recurrent network for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5277–5286, 2019.

[57] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 7942–7951, 2019.

[58] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5463–5474, 2021.

[59] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-DeepLab: Stand-Alone Axial-Attention for Panoptic Segmentation. In *ECCV*, 2020.

[60] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. *arXiv preprint arXiv:2202.03052*, 2022.

[61] Wenguan Wang, Tianfei Zhou, Fatih Porikli, David Crandall, and Luc Van Gool. A survey on deep learning technique for video segmentation. *arXiv preprint arXiv:2107.01153*, 2021.

[62] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8741–8750, 2021.

[63] Mark Weber, Jun Xie, Maxwell Collins, Yukun Zhu, Paul Voigtlaender, Hartwig Adam, Bradley Green, Andreas Geiger, Bastian Leibe, Daniel Cremers, et al. Step: Segmenting and tracking every pixel. *arXiv preprint arXiv:2102.11859*, 2021.

[64] Julia Wolleb, Robin Sandkühler, Florentin Bieder, Philippe Valmaggia, and Philippe C Cattin. Diffusion models for implicit image segmentation ensembles. *arXiv preprint arXiv:2112.03145*, 2021.

[65] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[66] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8818–8826, 2019.

[67] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5188–5197, 2019.

[68] Qihang Yu, Huiyu Wang, Dahun Kim, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Cmt-deeplab: Clustering mask transformers for panoptic segmentation. In *CVPR*, 2022.

[69] Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hatwig Adam, Alan Yuille, and Liang-Chieh Chen. k-means mask transformer, 2022.

[70] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-net: Towards unified image segmentation. *Advances in Neural Information Processing Systems*, 34:10326–10338, 2021.

[71] Zizhao Zhang, Han Zhang, Long Zhao, Ting Chen, Sercan Ö Arik, and Tomas Pfister. Nested hierarchical transformer: Towards accurate, data-efficient and interpretable visual understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3417–3425, 2022.