A. Algorithm details

For completeness, we include Algorithm 3 and 4 from Bit Diffusion [12] to provide more detailed implementations of functions in Algorithm 1 and 2.

Algorithm 3 Binary encoding and decoding algorithms (in Tensorflow).

```
import tensorflow as tf

def int2bit(x, n=8):
  # Convert integers into corresponding binary bits.
  x = tf.bitwise.right_shift(
      tf.expand_dims(x, -1), tf.range(n))
  x = tf.math.mod(x, 2)
  return x

def bit2int(x):
  # Convert binary bits into corresponding integers.
  x = tf.cast(x, tf.int32)
  n = x.shape[-1]
  x = tf.math.reduce_sum(x * (2 ** tf.range(n)), -1)
  return x
```

Algorithm 4 x_t estimation with DDIM updating rule.

```
def gamma(t, ns=0.0002, ds=0.00025):
  # A scheduling function based on cosine function.
  return numpy.cos(
      ((t + ns) / (1 + ds)) * numpy.pi / 2)**2

def ddim_step(x_t, x_pred, t_now, t_next):
  # Estimate x at t_next with DDIM updating rule.
  $\gamma_{\text{now}}$ = gamma(t_now)
  $\gamma_{\text{next}}$ = gamma(t_next)
  x_pred = clip(x_pred, -scale, scale)
  eps =$\frac{1}{\sqrt{1-\gamma_{\text{now}}}}$ * (x_t - $\sqrt{\gamma_{\text{now}}}$ * x_pred)
  x_next = $\sqrt{\gamma_{\text{next}}}$ * x_pred + $\sqrt{1-\gamma_{\text{next}}}$ * eps
  return x_next
```

B. More details on training and inference hyper-parameters

MS-COCO. For unconditional pretraining of the mask decoder, we train the model on mask resolution 128×128 for 800 epochs on MS-COCO with a batch size of 512 and scale jittering of strength $[1.0, 3.0]$. For both unconditional pretraining of the mask decoder, and image-conditional training of mask generation (encoder and decoder), we use input scaling of 0.1, loss weighting $p = 0.2$, learning rate of $1e^{-4}$, with EMA decay of 0.999.

Cityscapes. For fine-tuning on Cityscapes, we train for 800 epochs using a batch size of 16 and learning rate of $3e^{-5}$ linearly decayed to $3e^{-6}$ with no warmup and no scale jittering augmentation. Image size of 1024×2048 with mask size of 512×1024 are used. During eval, we use `td`$=1.5$ and resize the predicted mask to the full mask size of 1024×2048 using nearest neighbour resizing and filter out annotations with less than 80 pixels.

DAVIS. For fine-tuning on DAVIS, we train for 20k steps with batch size of 32, loss weighting of $p = 0.2$, a constant learning rate of $1e^{-5}$, EMA decay of 0.99, and scale jittering of strength $[0.7, 2]$. Image size of 512×1024 and mask size of 256×512 are used. For evaluation, as the dataset is quite small, we run inference 50 times for our model, and report the mean (the standard deviation for $\mathcal{J}\&\mathcal{F}$ is around 1.5).

method	backbone	params	PQ
Specialist approaches:			
Mask2Former [16]	ResNet-50 [22]	-	62.1
kMaX-DeepLab [69]	ResNet-50 [22]	56M	64.3
Panoptic-DeepLab [15]	Xception-71 [18]	47M	63.0
Axial-DeepLab [59]	Axial-ResNet-L [59]	45M	63.9
Axial-DeepLab [59]	Axial-ResNet-XL [59]	173M	64.4
CMT-DeepLab [68]	MaX-S [58]	-	64.6
Panoptic-DeepLab [15]	SWideRNet-(1,1,4.5) [9]	536M	66.4
Mask2Former [16]	Swin-B (W12) [41]	-	66.1
Mask2Former [16]	Swin-L (W12) [41]	-	66.6
kMaX-DeepLab [69]	MaX-S [58]	74M	66.4
kMaX-DeepLab [69]	ConvNeXt-B [42]	121M	68.0
kMaX-DeepLab [69]	ConvNeXt-L [42]	232M	68.4
Generalist approaches:			
Pix2Seq-$\mathcal{D}$ (steps=10)	ResNet-50 [22]	94.8M	62.2
Pix2Seq-$\mathcal{D}$ (steps=20)	ResNet-50 [22]	94.8M	63.2
Pix2Seq-$\mathcal{D}$ (steps=40)	ResNet-50 [22]	94.8M	63.4
Pix2Seq-$\mathcal{D}$ (steps=80)	ResNet-50 [22]	94.8M	64.0

Table 6: Cityscapes *val* set results.

	Mask size	
Image size	256×512	512×1024
512×1024	52.1	53.8
1024×2048	55.7	59.9

Table 7: PQ for various image sizes and panoptic mask sizes for Cityscapes *val* set. Model is trained for 100 epochs for ablation.

C. Results on Cityscape

Table 6 compares our results on Cityscapes *val* set with prior work. Our main results are with an image size of 1024×2048 and mask size of 512×1024. In Table 7 we show an ablation with varying image and mask sizes and we find that both a larger image size and a larger mask size are important.

D. Results on KITTI-STEP

In addition to video object segmentation on DAVIS, we also applied the same method to video panoptic segmentation on more recent KITTI-STEP dataset [63]. Training configurations remain the same as on DAVIS, but with image size 384x1248. For inference we use $td = 1.5$ and 10 sampling steps. Our preliminary results are shown in Table 8. Pix2seq-D achieved decent results (though behind the state-of-the-art), especially considering that minimal tuning or changes are done to apply Pix2seq-D to this task.

Method	STQ	SQ	AQ
Motion-DeepLab [63]	58.0	67.0	51.0
VPSNet [28]	56.0	61.0	52.0
TubeFormer-DeepLab-B1 [29]	70.0	76.8	63.8
Pix2Seq-D	61.6	61.9	61.3

Table 8: Comparison of results on KITTI-STEP. All methods listed have a Resnet-50 backbone.

Unlike most existing methods that do inference on video segments and stitch the segments in postprocessing, we do inference on the entire video in a streaming fashion. For the ease of experimentation, we only conduct inference on videos up to 400 frames with image size 384x1248. One video in the KITTI-STEP validation set exceeds 400 frames, and is therefore split into two videos during inference. We believe this only has a negligible impact on the overall metrics.

E. Extra Visualization

Figure 11 shows the inference trajectory of or model for two MS-COCO validation examples, we see that the model iteratively refines the panoptic mask outputs so that they become globally and locally consistent.

Figure 12 and 13 present more visualization of our model's predictions on MS-COCO validation set. Figure 14 and 15 present extra visualization of our model's predictions on Cityscapes and DAVIS validation sets, respectively.

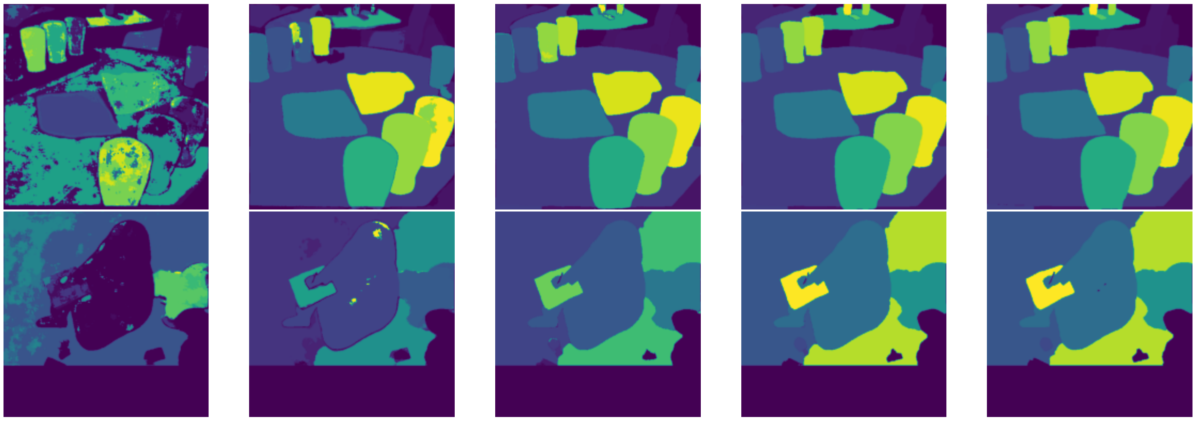

Figure 11: Inference trajectory. Predicted m_0 at different time steps (1, 2, 4, 8, 16) out of total 20 steps.

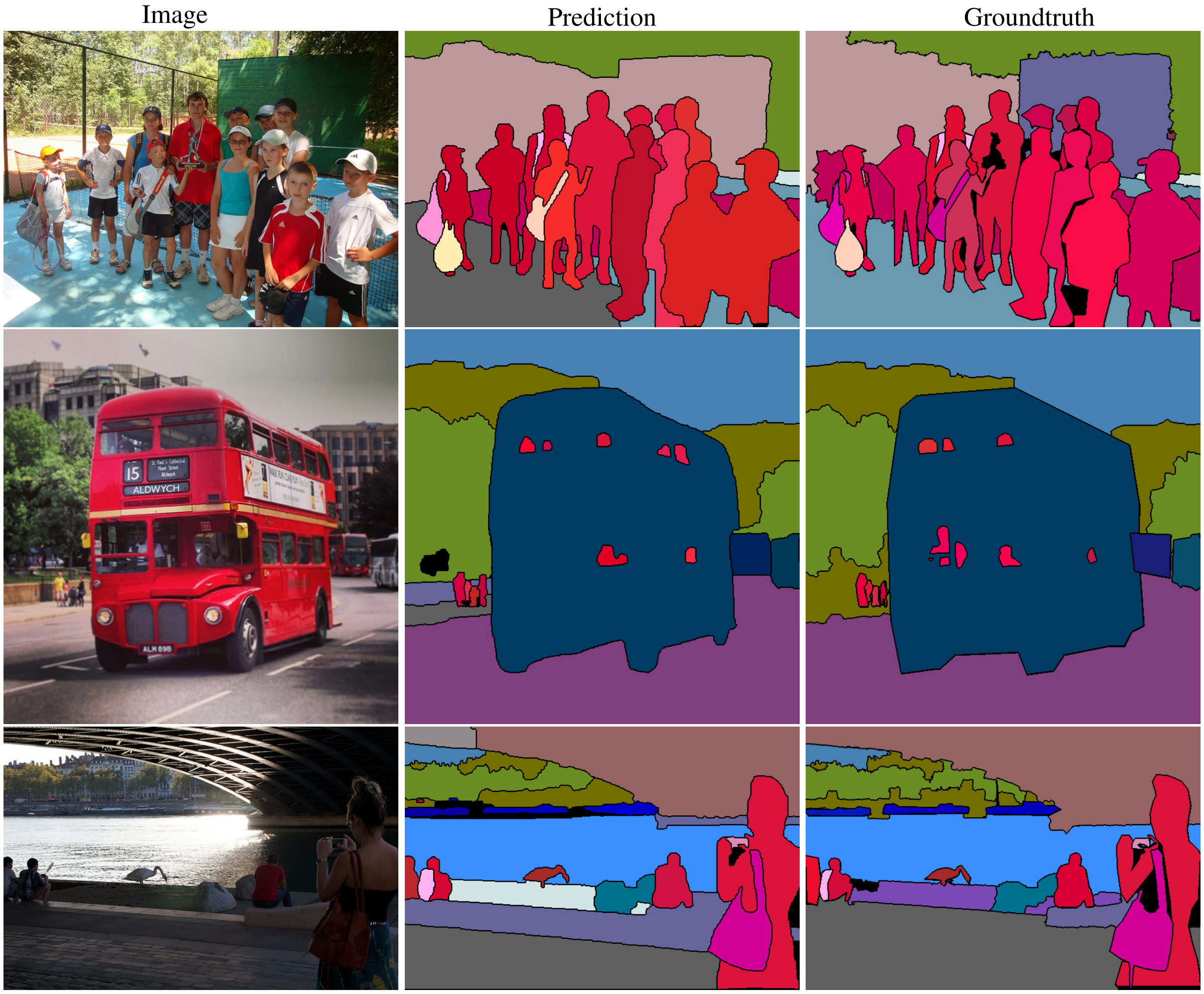

Figure 12: Predictions on MS-COCO *val* set.

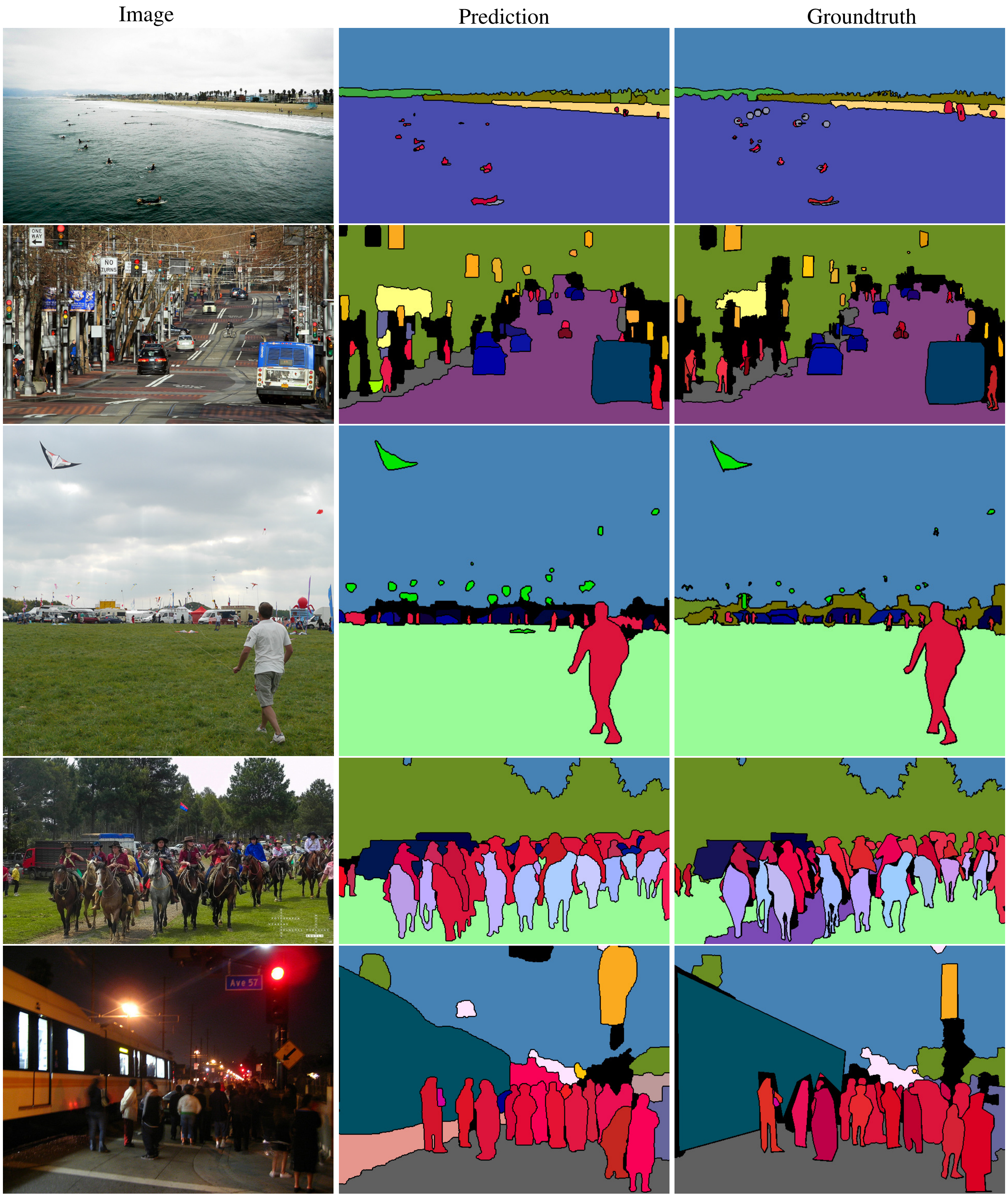

Figure 13: Predictions on MS-COCO *val* set.

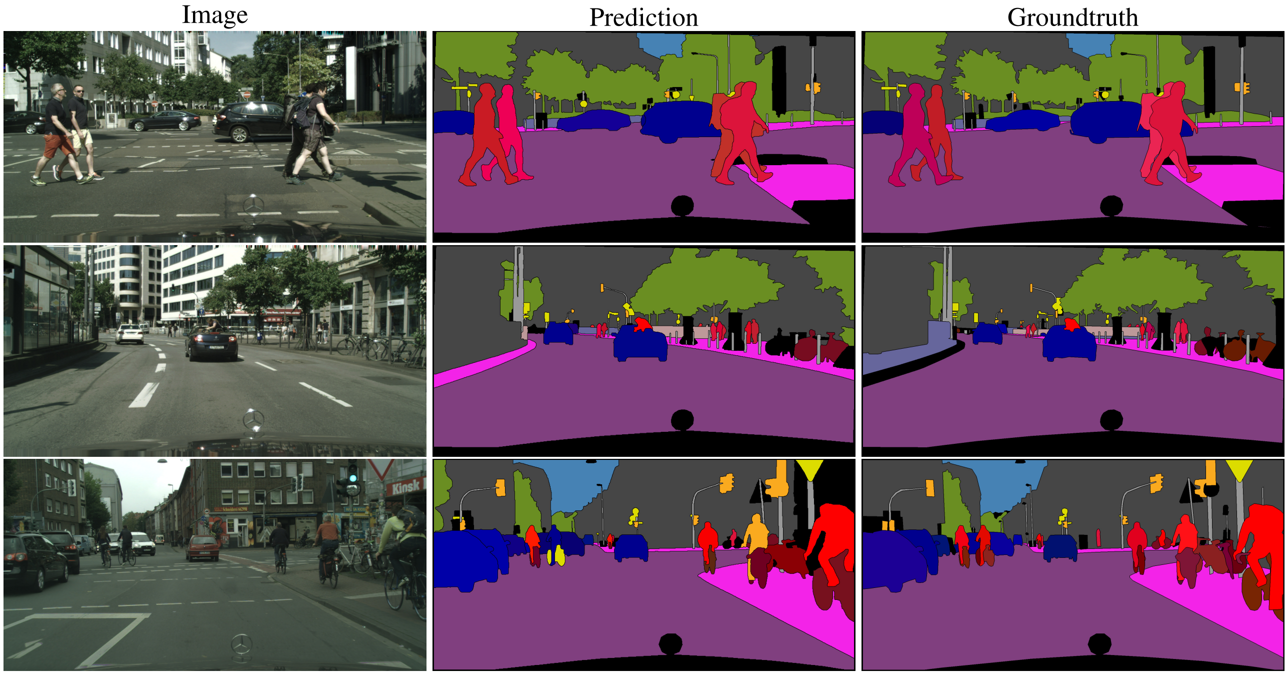

Figure 14: Predictions on Cityscapes *val* set.

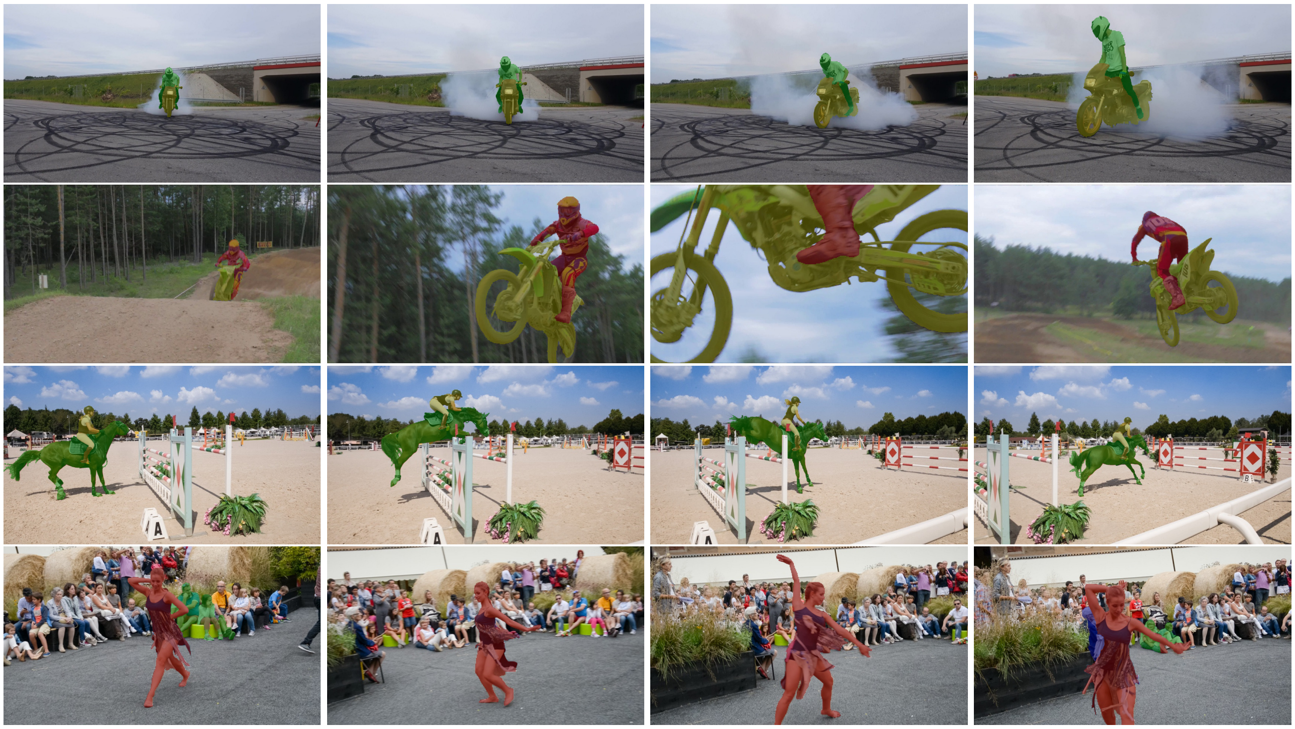

Figure 15: Predictions on DAVIS *val* set.