7. Supplementary material

7.1. Metrics details

The GED and HM-IoU metrics used in our work are computed as follows:

GED: Let p_m be the distribution over samples generated by a model and p_{gt} the distribution over possible ground-truth labels; the GED is computed as

$$\begin{aligned}\text{GED}(p_m, p_{gt}) =& 2\,\mathbb{E}_{s\sim p_m, \hat{s}\sim p_{gt}}[d(s,\hat{s})] - \mathbb{E}_{s,\hat{s}\sim p_{gt}}[d(s,\hat{s})] \\ & - \mathbb{E}_{s,\hat{s}\sim p_m}[d(s,\hat{s})], \end{aligned} \tag{19}$$

where the distance function $d(\cdot,\cdot) = 1 - \text{IoU}(\cdot,\cdot)$.

HM-IoU: Finds the optimal matching between ground truth and generated samples. Specifically, for n generated samples, the ground-truth samples are duplicated to n. Then, the HM-IoU is defined as the maximum IoU possible, given that every generated sample is matched with a unique ground-truth label, found by minimizing

$$\text{HM-IoU} = \min_X \sum_i \sum_j d(i,j) X_{i,j}, \tag{20}$$

where X is a boolean matrix that assigns every row to a unique column using $d(\cdot,\cdot) = 1 - \text{IoU}(\cdot,\cdot)$.

7.2. Sample diversity

Sample diversity is the expected distance between generated samples, *i.e.*, $\mathbb{E}_{s,\hat{s}\sim p_m}[d(s,\hat{s})]$, which corresponds to the last term of GED in Eq. (19). We report the sample diversity for 16, 32, 50, and 100 samples for both LIDC splits in Tab. 4 and Tab. 5.

	LIDCv1			
Method	Div_{16}	Div_{32}	Div_{50}	Div_{100}
CCDM	$0.491_{\pm 0.001}$	$0.509_{\pm 0.001}$	$0.515_{\pm 0.002}$	$0.519_{\pm 0.002}$

Table 4: Sample diversity for our method on LIDCv1.

	LIDCv2			
Method	Div_{16}	Div_{32}	Div_{50}	Div_{100}
CCDM	$0.487_{\pm 0.003}$	$0.503_{\pm 0.003}$	$0.509_{\pm 0.003}$	$0.515_{\pm 0.002}$

Table 5: Sample diversity for our method on LIDCv2.

7.3. Model size

While our 9M CCDM as reported in Tab. 1 is of comparable size to most other baselines, we show in Tab. 6 that by increasing the size of our CCDM from 9M to 41M, we get an increase in performance across all six metrics computed on LIDCv1. Additionally, the CCDM seems to benefit more from the increase in size than MoSE [15]. While we already outperform the other baselines with our 9M model, this result suggests that we can improve the performance even further by using larger models.

		LIDCv1					
Method	#params	GED_{16}	GED_{32}	GED_{50}	GED_{100}	HM-IoU_{16}	HM-IoU_{32}
MoSE [15]	9m	0.219	-	0.195	0.190	0.620	-
MoSE [15]	42m	0.218	-	0.195	0.189	0.624	-
CCDM	9m	0.212	0.194	0.187	0.183	0.623	0.631
CCDM	41m	0.207	0.189	0.182	0.177	0.629	0.636

Table 6: Performance of CCDM and MoSE on LIDCv1 with different model sizes.

7.4. Training settings of baselines on Cityscapes

On Cityscapes, all baselines were trained for 500 epochs using the optimizer, learning rate schedule, and weight decay (denoted by w_d) reported in their original publications. Tab. 7 details these settings for each case. All models are trained using a cross-entropy loss.

Method		Settings				
Arch.	Backbone	Lr	Decay	w_d	Batch Size	Optim
HRNet [47]	w48v2	10^{-2}	polynomial	5×10^{-5}	32	sgd
DeepLabv3 [7]	ResNet50/101	10^{-2}	polynomial	5×10^{-5}	32	sgd
UPerNet [51]	ResNet101	10^{-2}	polynomial	5×10^{-5}	32	sgd
UPerNet [32]	Swin-T	10^{-4}	warmup+linear	10^{-2}	32	AdamW

Table 7: Training settings of baselines on Cityscapes.

7.5. Additional comparisons on Cityscapes

Method			mIoU	
Architecture	Backbone	#params	128×256	256×512
UNet (CE) [13]	-	30m	48.7	61.0
CCDM (ours)	-			
samples=1		30m	53.2	60.3
samples=5		30m	55.4	62.0
samples=10		30m	56.2	62.4
UNet (CE) [13]	Dino ViT-S (†)	30m + 20M	53.4	63.2
CCDM (ours)	Dino ViT-S (†)			
samples=1		30m + 20M	55.5	64.0
samples=5		30m + 20M	<u>56.9</u>	<u>65.4</u>
samples=10		30m + 20M	**57.3**	**65.8**

Table 8: Comparison of our method to UNet and UNet-Dino, trained with standard Cross-Entropy (CE) loss, on Cityscapes-val. **Bold** and <u>underlined</u> indicate best and second best per column, respectively. (†) indicates self-supervised pretraining of the backbone. Gray indicates pretrained, non-finetuned parameters.

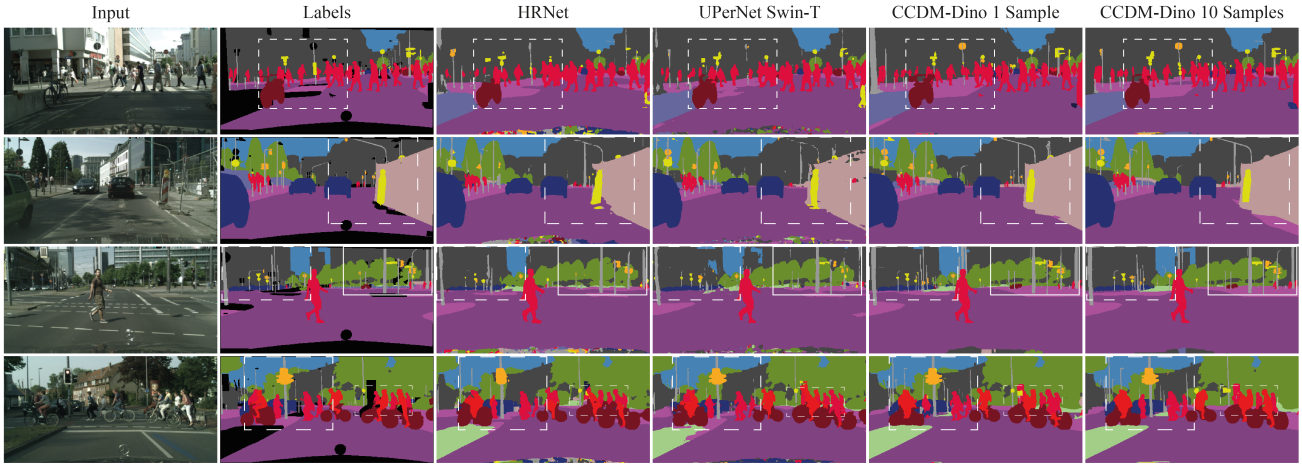

Figure 6: Qualitative comparisons of our method to competitive baselines on Cityscapes validation set.

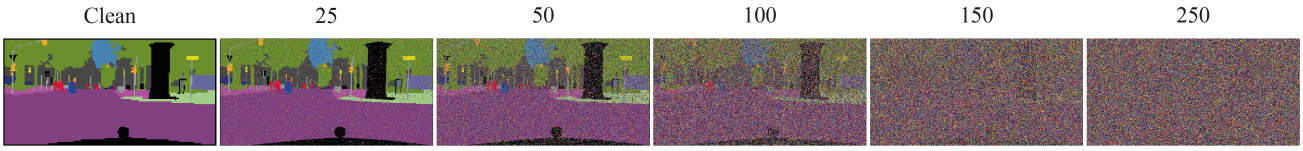

Figure 7: Visualization of the forward diffusion process at different time steps.

We evaluate the gains of CCDMs with respect to their backbone architectures when used as standalone segmentation models. To this end, we compare the performance of our CCDM trained as defined in Alg. 1 and the UNet trained with a standard cross-entropy loss, both on the Cityscapes dataset. Similarly, we compare CCDM-Dino to its standalone backbone architecture DinoViT-S. In all cases, we adopt the same training settings as our method, namely, 800 epochs, linearly decayed learning rate, batch size of 32 at 128×256 and 16 at 256×512. As shown in Tab. 8, CCDM and CCDM-Dino outperform their respective standalone architectures.

We also provide additional qualitative comparisons of our method to competitive baselines in Fig. 6. Finally, Fig. 7 shows an example of the evolution of a Cityscapes label map under the forward diffusion process described by Eq. (4).