Appendix

S1. Implementation Details

In this section, we provide more implementation details in our work.

Training details. For CycleGAN and Pix2pix models, we use batch size of 32 for teacher and batch size of 80 for student, while for GauGAN, the batch size is set to 16 for both. For each model and each dataset, we apply the same training epochs for teacher and student networks. The learning rate for both generators and discriminators are set as 0.0002 for all datasets and models. More detailed training hyperparameters are summarized in Table S1. For the layers used for knowledge distillation between teacher and student networks, we follow the same strategy as Li *et al*. [36]. Specifically, for Pix2pix and CycleGAN models, the 9 residual blocks are divided into 3 groups, each with three consecutive layers, and knowledge is distilled upon the four activations from each end layer of these three groups. For GauGAN models, knowledge distillation is applied on the output activations of 3 from the total 7 SPADE blocks, including the first, third and fifth ones.

More details for normalization layers. We find that instance normalization [69] without tracking running statistics is critical for dataset Horse→Zebra to achieve good performance on the student model, and for dataset Zebra→Horse, synchronized batch normalization with tracked running statistics gives better performance. For the other datasets batch normalization [28] with tracked running statistics is better. Normalization layers without track running statistics introduce extra computation cost, and we take this into account for our calculation of MACs during pruning. Moreover, for GauGAN, we use synchronized batch normalization as suggested by previous work [58, 67], and remove the spectral norm [55] as we find it does not have much impact on the model performance.

Network details for GauGAN. For GauGAN, we find it is sufficient for each spade residual block to keep only the first SPADE module in the main body while replace the second one as well as the one in the shortcut by synchronized batch normalization layer. This saves computation cost by a large extent. Besides, we use learnable weights for the second synchronized block for the purpose of pruning. These weights do not introduce extra computation cost, as the running statistics are estimated from training data and not recalculated during inference, enabling fusing normalization layers into the convolution layers. Further, we replace the three convolution layers in the SPADE module by our proposed inception-based residual block (IncResBlock), with normalization layers included for pruning. The details for the architecture are illustrated in Figure S1. We name our SPADE module as IncSPADE and SPADE residual block as IncSPADE ResBlk.

To prune the input channel for each model, we add an extra normalization layer (synchronized batch normalization) with learnable weights after the first fully-connected layer, and prune its channels together with other normalizations using our pruning algorithm described in the Section 3.2 of the main paper. During pruning, we keep the ratio of input channels between different layers as the original model, and the lower bound for the first layer (which has the largest number of channels) is determined by that for the last layer multiplied by the channel ratio, so that all channels are above the bound and the channel ratio is unchanged.

S2. Ablation Analysis of Knowledge Distillation

Here we show the ablation analysis for knowledge distillation methods. We use our searching method to find a student architecture on Pix2pix task using the Cityscapes dataset, and compare student training without knowledge distillation, with MSE distillation as in [36], and the similarity-based distillation we proposed. The results are summarized in Tab. S2, where *w/o Distillation* denotes training the student without distillation, and *w/ MSE; Loss Weight* 0.5 and *w/ MSE; Loss Weight* 1.0 denotes MSE distillation with weight 0.5 and 1.0, respectively. We find that distillation indeed improves performance, and our distillation method, which employs GKA to maximize feature similarity, is better than MSE on transferring knowledge from teacher to student via intermediate features.

S3. More Qualitative Results

We show more qualitative results for CycleGAN on Horse→Zebra and Zebra→Horse, Pix2pix on Map→Aerial photo, as well as GauGAN on Cityscapes in Figs. S2, S3, S4, and S5, respectively.

Table S1: Hyper-parameter setting for teacher and student training.

Model	Dataset	Training Epochs		λ_{distill}	λ_{recon}	λ_{fm}	GAN Loss	ngf	ndf
		Const	Decay					Teacher	
CycleGAN	Horse→Zebra	500	500	1	5	-	LSGAN	64	64
	Zebra→Horse	500	500	0.1	5	-	LSGAN	64	64
Pix2pix	Cityscapes	500	750	0.5	100	-	Hinge	64	128
	Map→Aerial photo	500	1000	1.3	100	-	Hinge	64	128
GauGAN	Cityscapes	100	100	0.5	10	10	Hinge	64	64

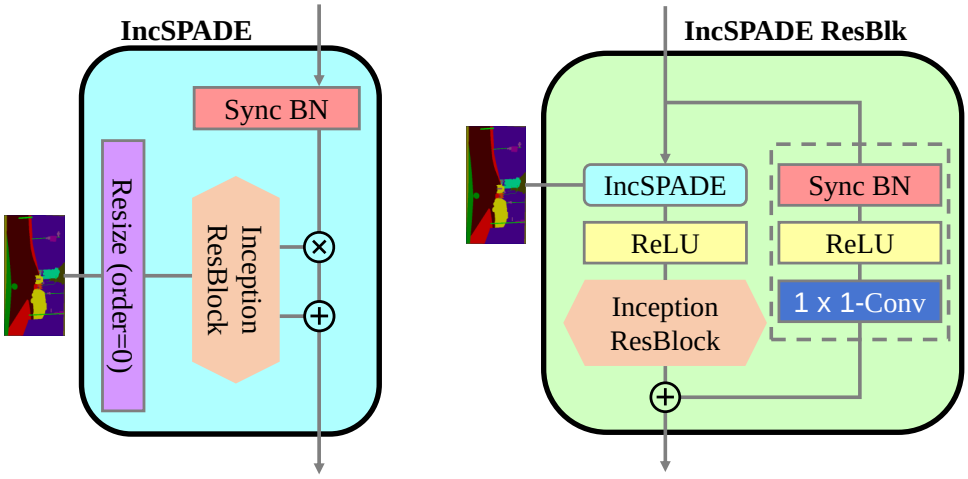

Figure S1: SPADE normalization module (IncSPADE, *left*) and SPADE residual block (IncSPADE ResBlk, *right*) with the proposed Inception Resblock (orange hexagon). Note that the optional last normalization layer and residual connection are not applied in the Inception Resblocks that are used in IncSPADE and IncSPADE ResBlk.

Table S2: Analysis of knowledge distillation methods on Cityscapes dataset with the Pix2pix setting. Our methods (GKA) achieves the best result.

Method	mIoU↑
w/o Distillation	39.39
w/ MSE; Loss Weight 0.5	39.83
w/ MSE; Loss Weight 1.0	39.76
Ours	**42.53**

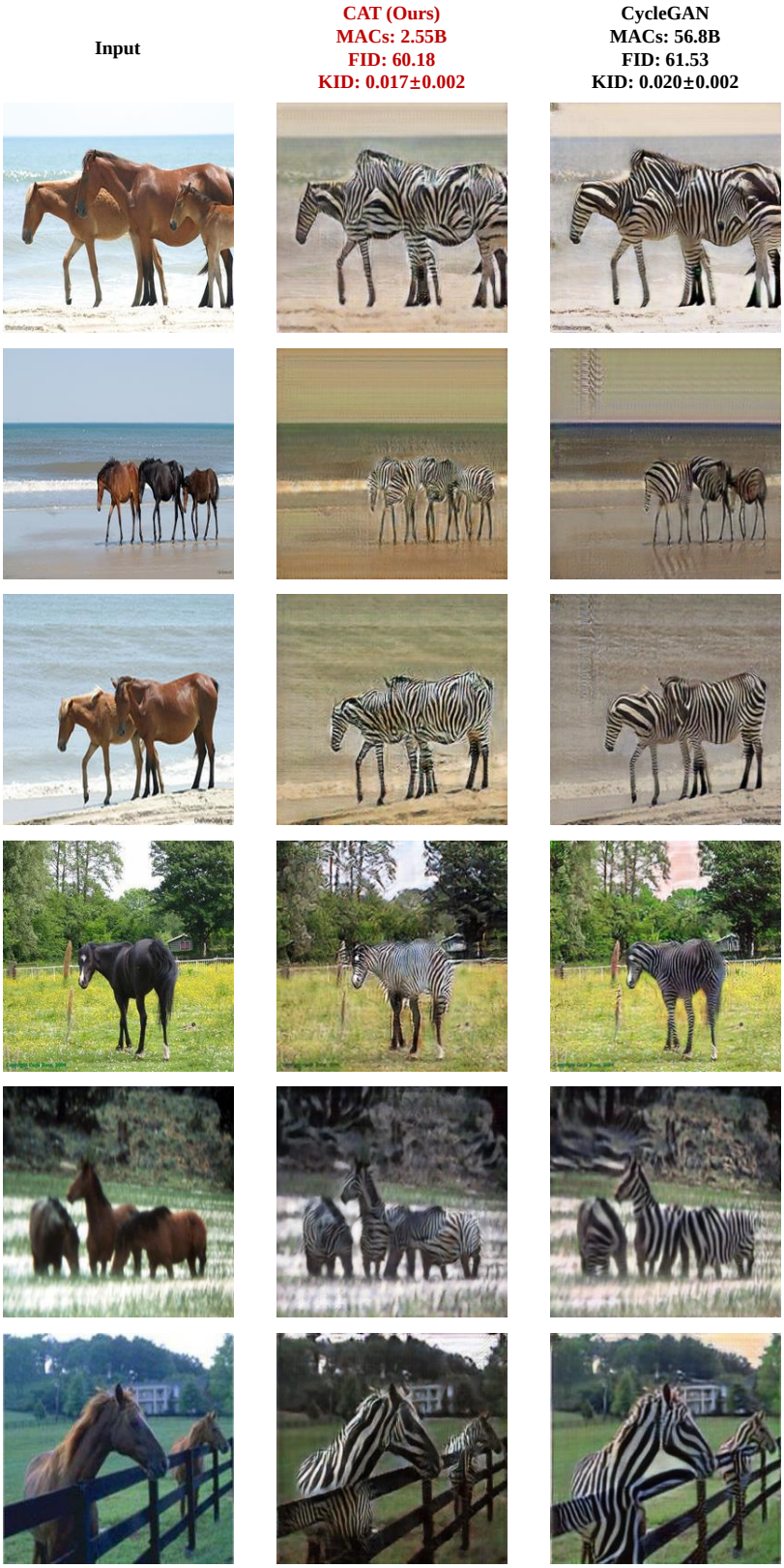

Figure S2: More results on Horse→Zebra dataset. Compared with original CycleGAN, our model has much reduced MACs and can generate images with higher fidelity (lower FID).

Figure S3: Results on Zebra→Horse dataset. Compared with original CycleGAN, our model has much reduced MACs and can generate images with higher fidelity (lower FID).

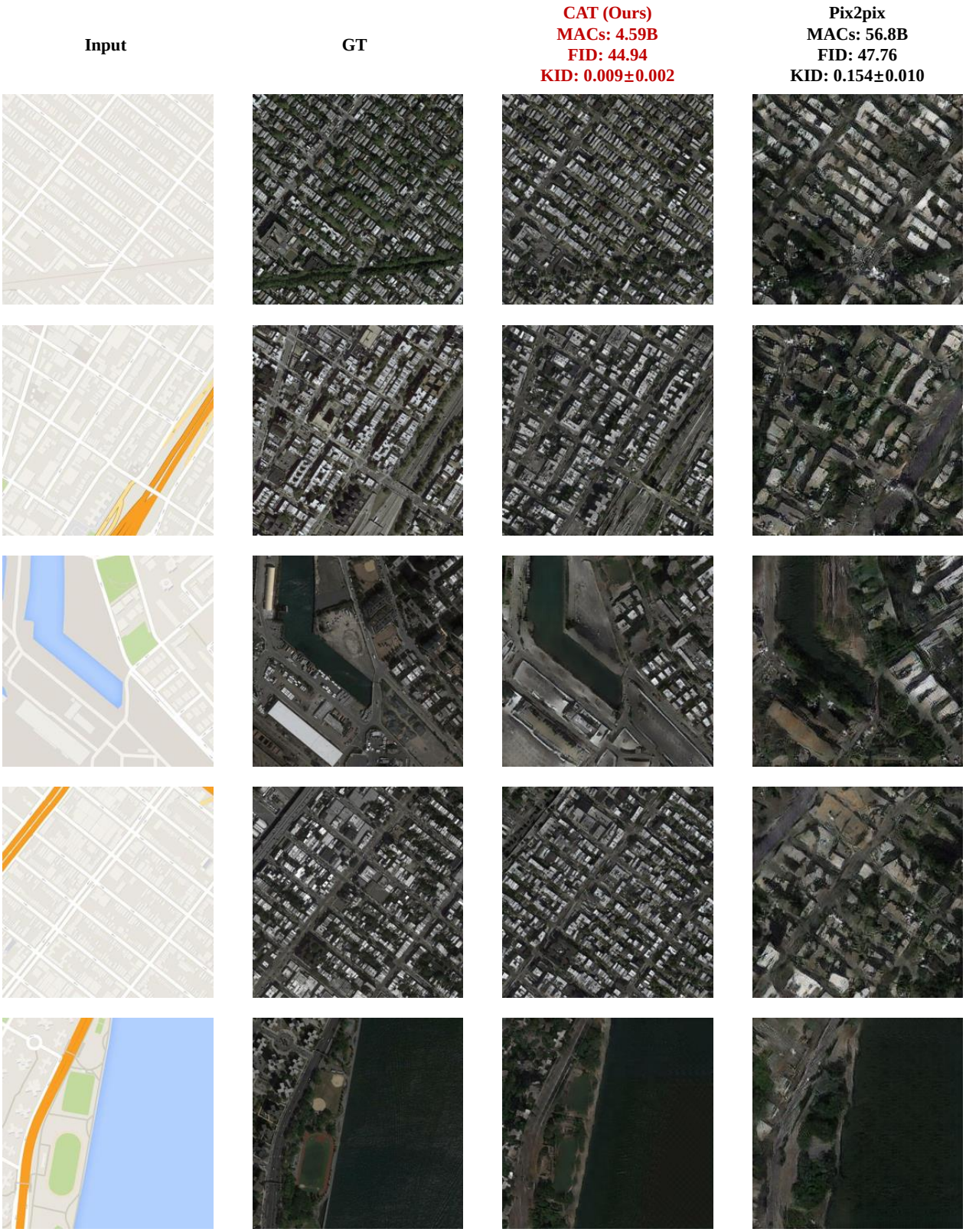

Figure S4: More results on Map→Aerial photo dataset. Compared with original Pix2pix, our model has much reduced MACs and can generate images with higher fidelity (lower FID).

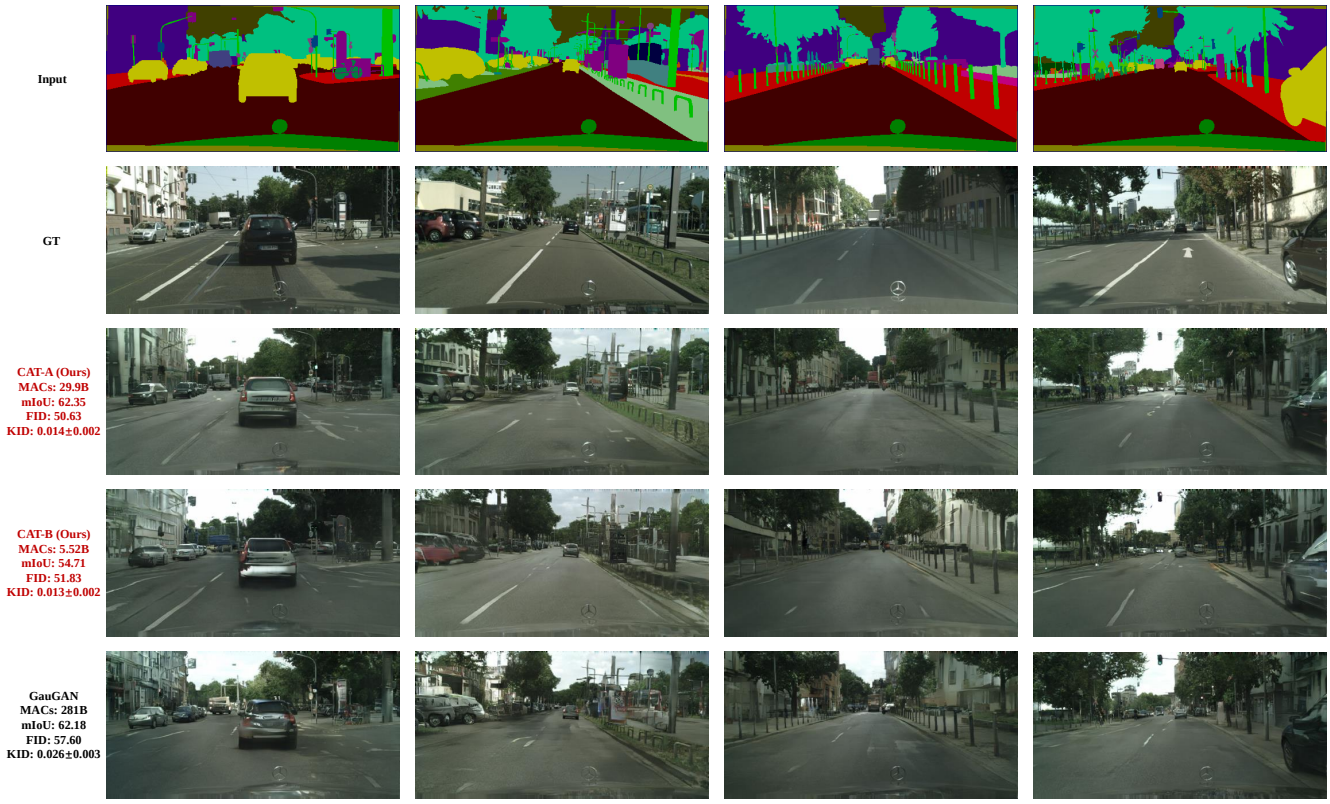

Figure S5: More qualitative results on Cityscapes dataset. Images generated by our compressed model (CAT-A, third row) have higher mIoU and lower FID than the original GauGAN model (fifth row), even with much reduced computational cost. For our CAT-B model (fourth row, $50.9\times$ compressed than GauGAN), although it has lower mIoU, the CAT-B model can synthesize higher fidelity images (lower FID) than GauGAN.