Don't Look into the Dark: Latent Codes for Pluralistic Image Inpainting

Supplementary Material

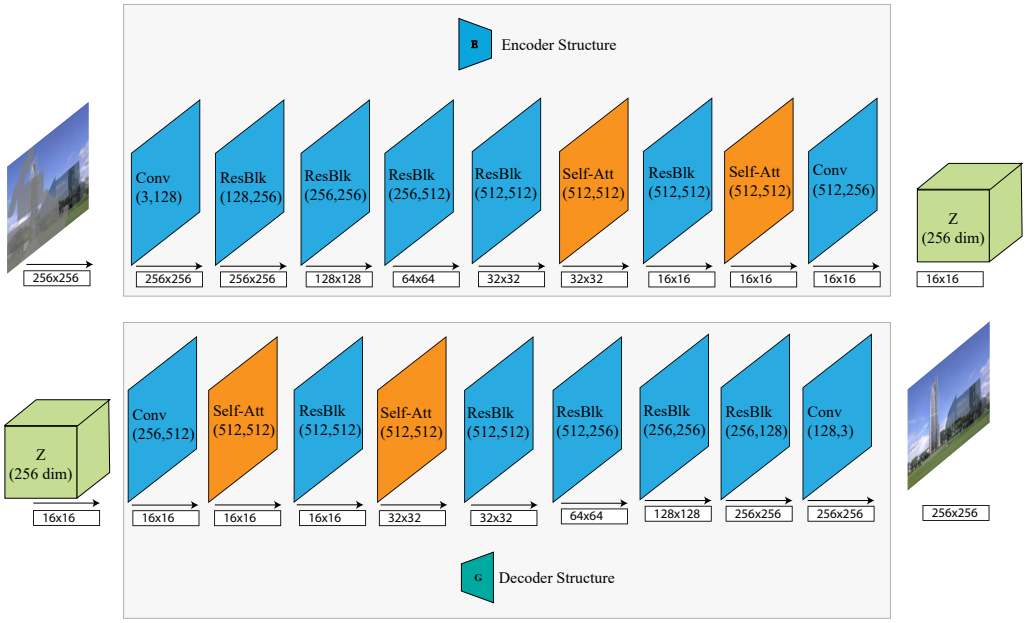

Figure 8. Detailed network structures for the encoder and decoder. Numbers within each feature map (e.g. (3,128)) denote the input and output channels. Numbers below each feature map (e.g. 256x256) denotes the size of the tensor.

6. Implementation Details

The network structures and all hyper-parameter settings are identical for both the Places model and the CelebA-HQ model. Our training pipeline is divided into three separate stages: the training of the encoder, the transformer, and the decoder. The free-form random masks [23] are used during the training of the encoder and the decoder. For the pre-trained VQGAN model, we use a quantization of $N = 1024$ tokens, each with a 256-channel embedding. All models are trained on 3 NVidia V100 GPUs with a batch size of 8. The Places model is trained for 20 epochs, and the CelebA-HQ model is trained for 200 epochs. The inference time of our model is around 0.4 seconds for an image on a single NVidia V100 GPU regardless of the mask size.

Figure 8 shows the detailed network structure designs for the encoders and decoder used in our method. Given input channel c_{in} and output channel c_{out}, "ResBlk" is a resnet block composed of two convolution layers and a skip connection. The two convolutions have weight matrices $W_1 \in \mathbb{R}^{3\times3\times c_{in}\times c_{out}}$ and $W_2 \in \mathbb{R}^{3\times3\times c_{out}\times c_{out}}$ respectively. Layer normalization is applied before and after the first convolution. The restrictive encoder and partial encoder described in the method part (Section 3) replace all convolutions with the restrictive convolution and the partical convolution respectively. In addition, self-attention layers (denoted by "Self-Att" in the figure) are added to process the features at 32x32 and 16x16 resolution.

The transformer model described in Section 3.2 is designed based on the minGPT transformer model*, with token embedding and positional embedding of 1408 channel and 40 layers of 16-head attention layers. During training, we have applied attention dropout and embedding dropout both with a 10% probabilities.

7. Detailed Loss Function

The loss functions used in training the decoder network described in Section 3.3 involve an adversarial loss function [12]:

$$\mathcal{L}_G = -\mathbb{E}_{\hat{x}}[\log(D(\hat{x})], \tag{10}$$

$$\mathcal{L}_D = -\mathbb{E}_{x}[\log(D(x)] - \mathbb{E}_{\hat{x}}[\log(1 - D(\hat{x})], \tag{11}$$

where x and $\hat{x}$ are a pair of real and fake samples, G, D are the generator and the discriminator. We additional use the R1 regularization [30] of the form:

*see https://github.com/karpathy/minGPT

$$R_1 = \mathbb{E}_x \|\nabla D(x)\|. \tag{12}$$

The LPIPS reconstruction loss function [47] is formulated as

$$\mathcal{L}_P = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \|w_l \odot (\hat{y}^l_{hw} - \hat{y}^l_{0hw}\|_2^2, \tag{13}$$

which compute the L2 distance between the layer activation of a pretrained VGG network [33] at each layer l.

8. Additional Visual Results

Please refer to Figure 9, 10, 11, 12 for additional visual comparisons to the baseline methods on both the Places [50] and the CelebA-HQ [18] dataset.

9. Further ablation analysis

Ablation study on the sampling function. Figure 13 provides more visual examples that compare between inpainting results under different sampling temperature t and annealing factor s (described in Section 3.2). As suggested by the examples, lower temperature in the sampling function leads to less diverse results and homogeneous textures in the synthesized areas (particular in natural scene inpainting). Higher temperature, on the other hand, leads to more diverse results at the expanse of visual coherence. The bottom three rows of Figure 13 best illustrates this phenomenon: the small mustache region left in the original masked image should naturally encourage the inpainting algorithm to synthesize a male face. However, since setting a higher temperature encourages the sampler to sample random latent codes in the early steps, female faces are synthesized regardless of the present evidence.

Ablation study on the restrictive encoder. We further study the effectiveness of the restrictive encoder by comparing it to a miracle encoder that has access to the original complete image. Specifically, while our restrictive encoder takes as input the masked image and produces latent code by $z_0 = E(MX)$, the miracle encoder encodes the complete image as input and masks the latent codes with a downsampled mask, with $z_1 = \hat{M}E(X)$. Figure 14 provides a visual comparison between the two encoders. For the specific example in the figure, z_0 and z_1 only share 24.8% of the encoded latent codes, as the miracle encoder manages to encode the image differently with access to the complete image. The quality of the inpainting results, however, show little difference, although the restrictive encoder has to infer latent codes with far less information. Quantitative evaluation in Table 3 provides a comparison between the performance of the restrictive encoder and the miracle encoder when used in the full inpainting pipeline, where we found that the designed restrictive encoder can be nearly as effective as one that has full access to the complete images. This indicates that meaningful inductive bias has been learnt by the encoder in the training process.

Methods	Places (256×256)		
	FID↓		Diversity↑
	Small Mask	Large Mask	Box
Restrictive	1.02	2.82	0.29±0.06
Miracle	0.93	2.71	0.29±0.06

Table 3. Comparisons of FID and diversity scores between the restrictive encoder and the miracle encoder.

10. Time and Memory Complexity

Our model takes 314ms to generate an 256×256 inpainted image, with 25ms on encoding, 61ms on decoding and 228ms on predicting tokens, which is similar to the speed of the generative-transformer-based method MaskGIT [4] (298ms), but slower than the one-pass inpainting methods such as LaMa [34] (86ms) and MAT [22] (83ms). Inference with our model takes 5893Mb of GPU memory to process a single image.

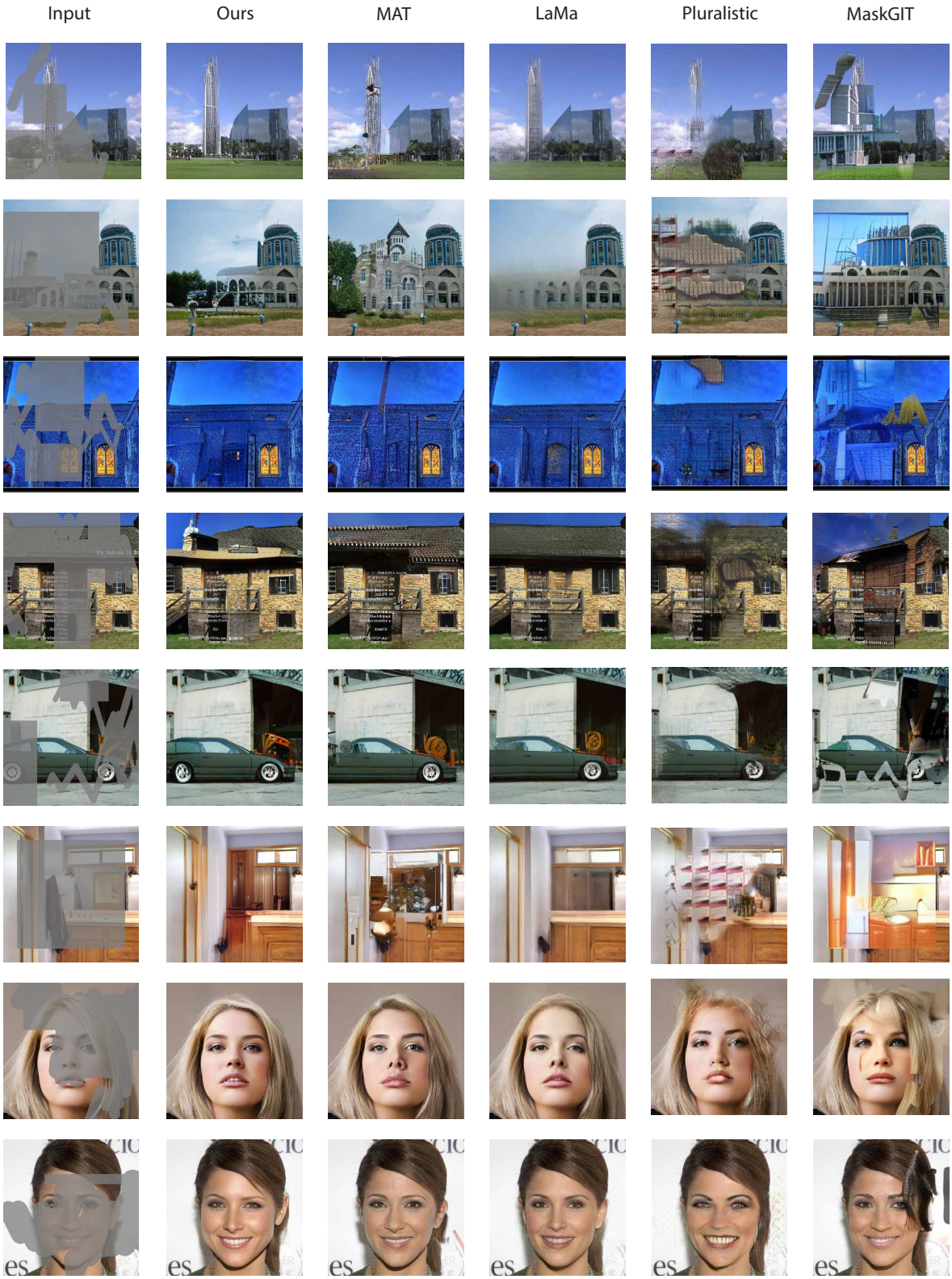

Figure 9. Further visual examples of inpaining under the large mask setting, compared to the baseline methods.

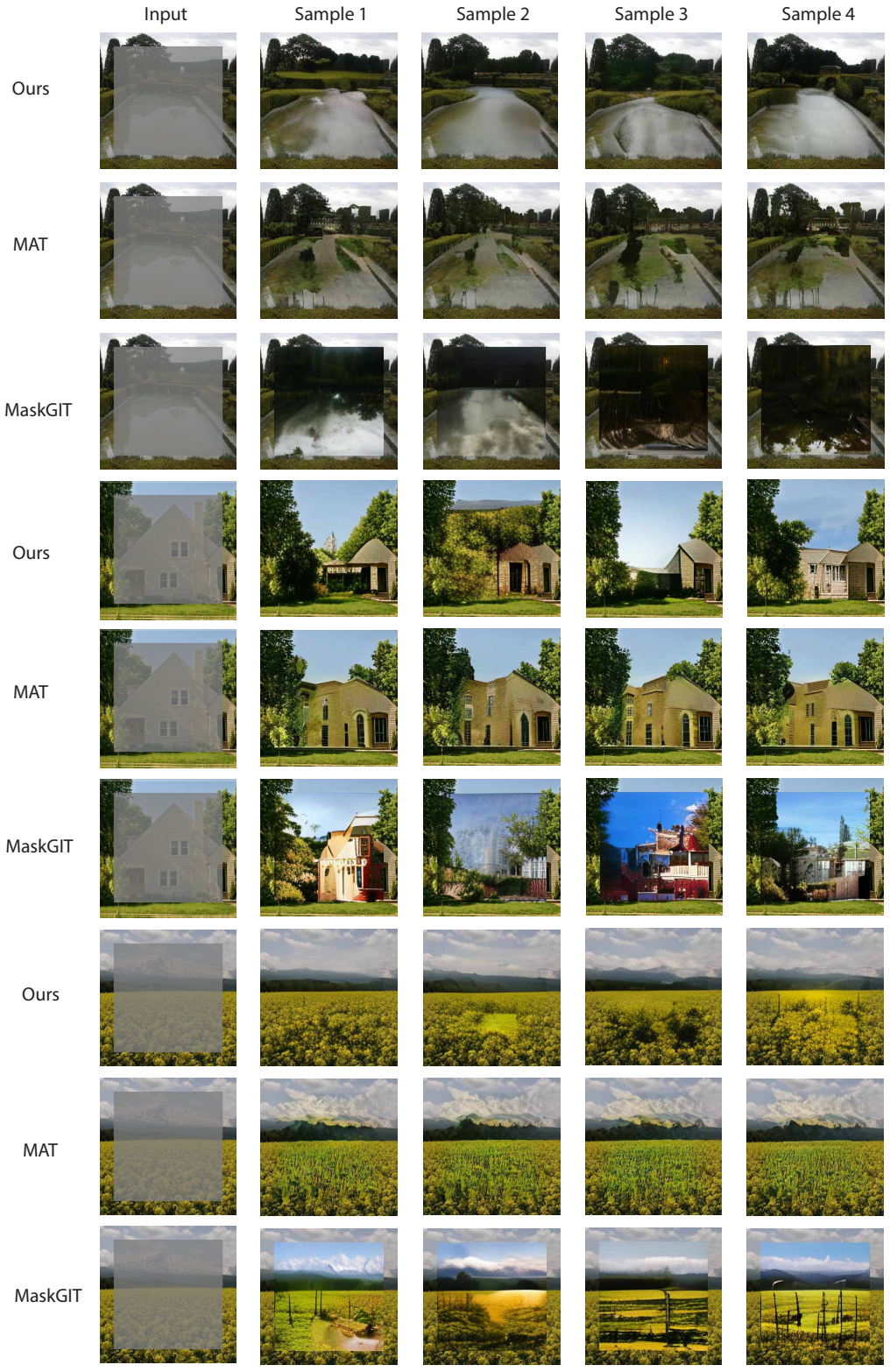

Figure 10. Further visual examples of pluralistic inpainting on the Places Dataset [50], compared to the baseline methods.

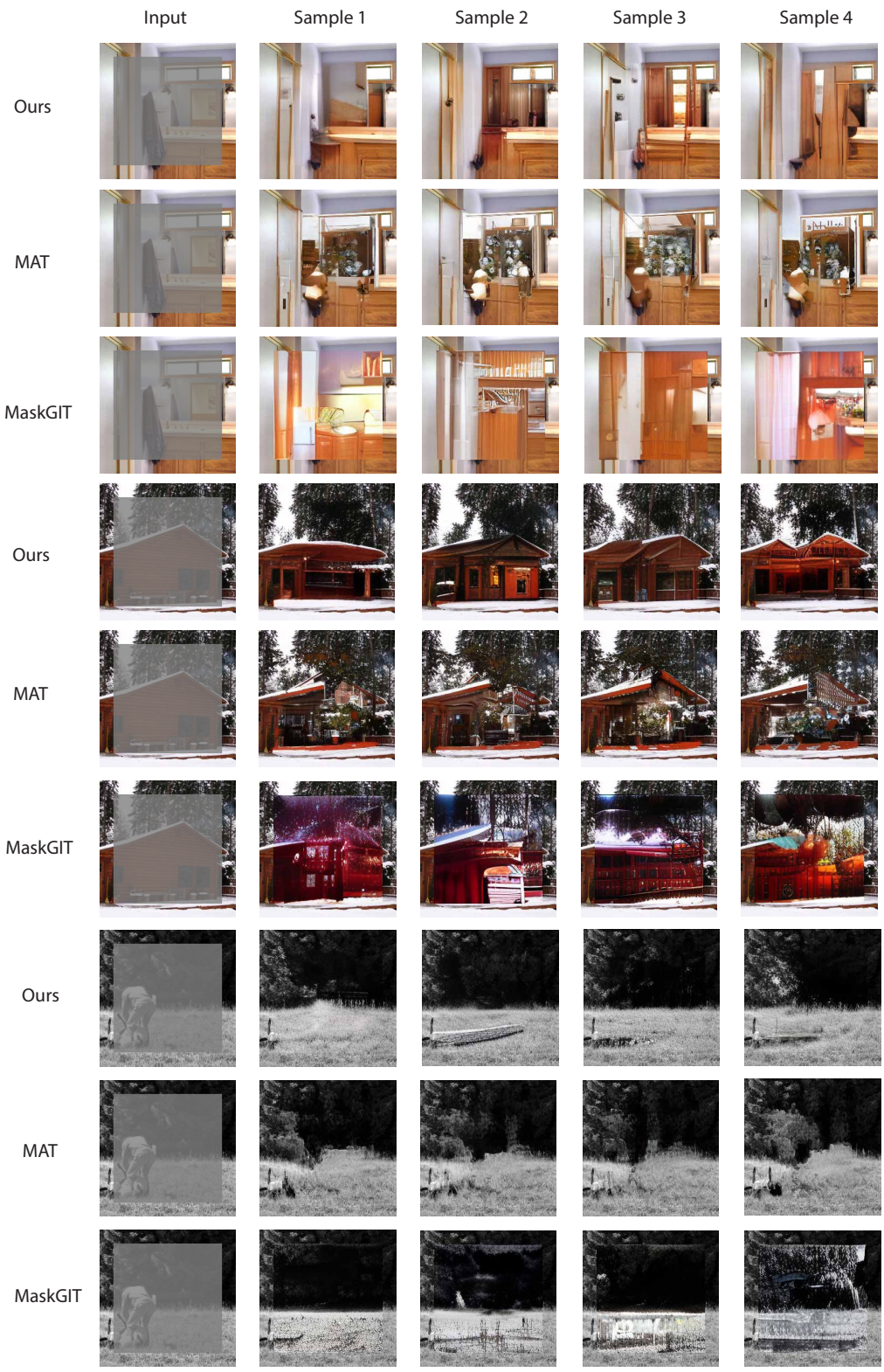

Figure 11. Further visual examples of pluralistic inpainting on the Places Dataset [50], compared to the baseline methods.

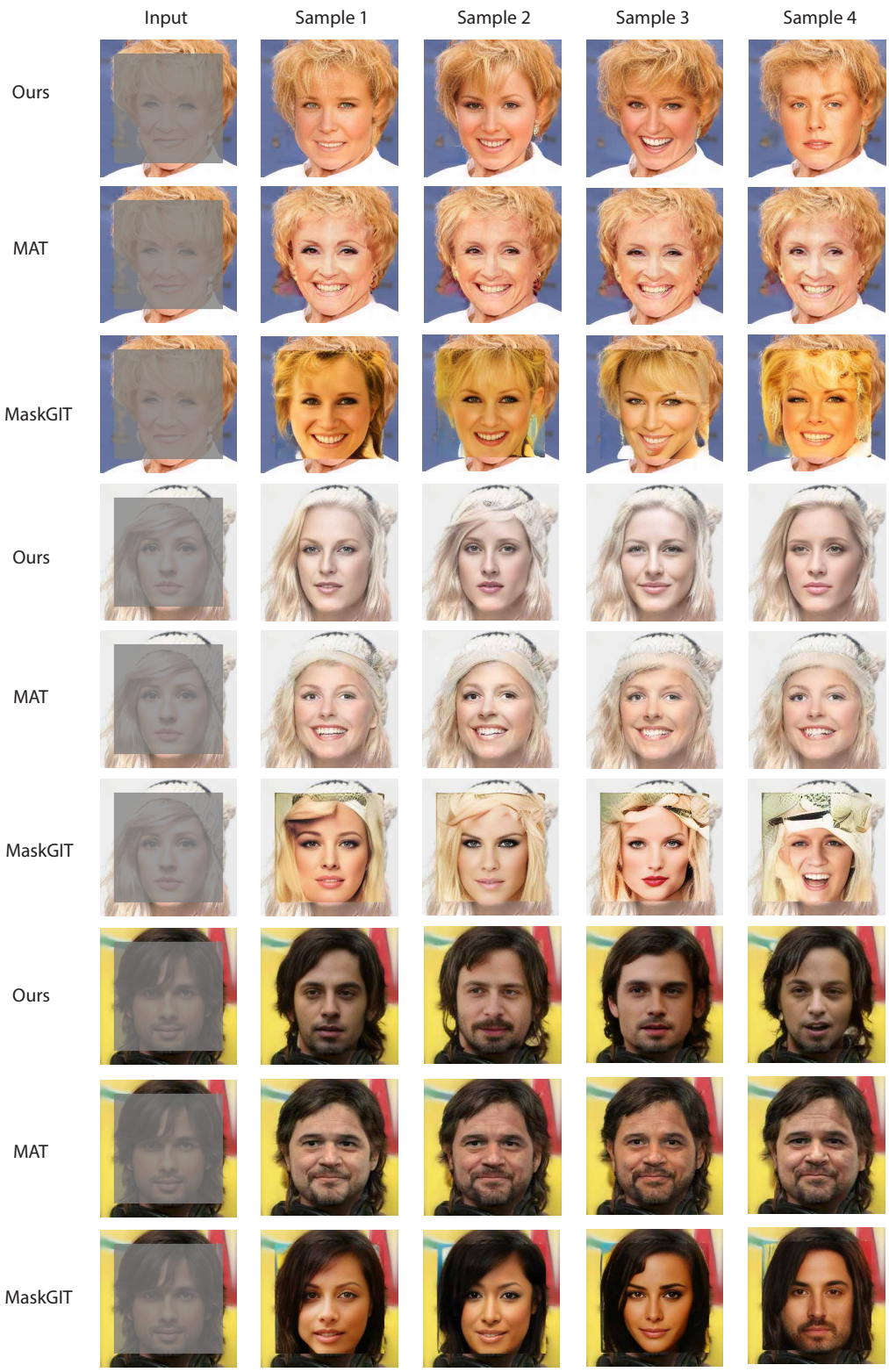

Figure 12. Further visual examples of pluralistic inpainting on the CelebA-HQ Dataset [18], compared to the baseline methods.

Figure 13. Further visual examples of pluralistic inpainting with respect to different sampling temperature t and annealing factor s.

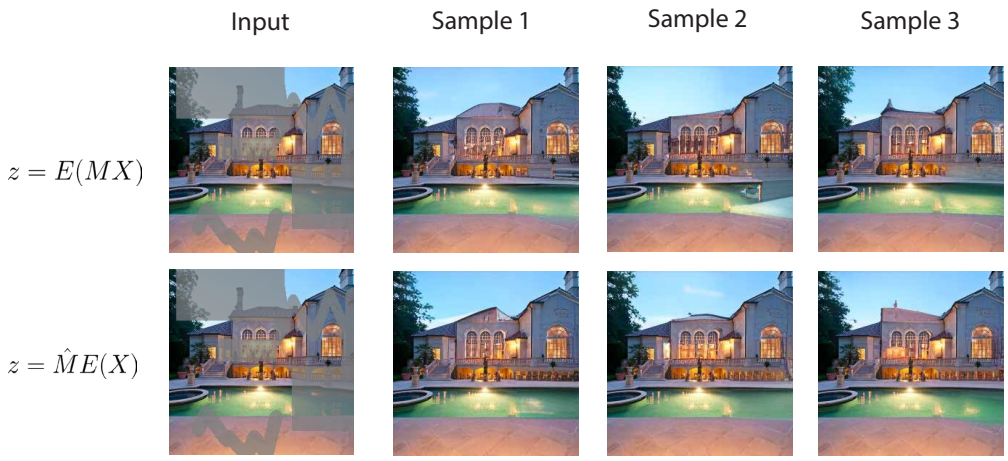

Figure 14. Visual comparison between inpainting with the restrictive encoder and a miracle encoder.