Appendix A. Network Architectures

Let $C_k$ denote a convolution layer with $k$ filters, $C_k^2$ a convolution layer with stride 2, $U$ a (2x) spatial upsampling layer, $D_k$ a dense layer with $k$ filters and $F$ a flattening layer. All convolution layers use a kernel size of 3 except for the first convolution of the encoder of each UNet which uses a kernel size of 7. Convolutions and dense layers are followed by a Leaky ReLU activation unless specified otherwise. All input volumes to the following networks have spatial resolution of $256 \times 256$.

A.1. Source Image Segmentation Network

Input: $[I_s, p_s] \in \mathbb{R}^{256 \times 256 \times 17}$
Output: $M_s \in \mathbb{R}^{256 \times 256 \times 11}$

Encoder:
C64, C64, C128, C128, C128, C128, C128, C128, C128, C128
Decoder:
C128, U, C128, U, C128, U, C128, U, C128, U, C64, C11

Skip layers concatenate output volumes (after activation) of layers 1, 3, 5, 7, 9 of the encoder to the outputs volumes (after activation) of layers 3, 5, 7, 9, 11 of the decoder. The final convolution of the decoder, C11 is followed by a linear activation.

A.2. Background Synthesis Network

Input: $[I_s, p_s] \in \mathbb{R}^{256 \times 256 \times 17}$
Output: $y_{bg} \in \mathbb{R}^{256 \times 256 \times 3}$

This architecture is identical to the source image segmentation network, except that the final convolution, C11, is replaced by C3 and is followed by a $tanh$ activation.

A.3. Foreground Synthesis Network

Input: $[W, p_t] \in \mathbb{R}^{256 \times 256 \times 44}$
Output: $y_{fg} \in \mathbb{R}^{256 \times 256 \times 3}$, $M_t \in \mathbb{R}^{256 \times 256 \times 1}$

Encoder:
C128, C128, C128, C128, C256, C256, C256, C256, C256, C256
Decoder:
C256, U, C256, U, C256, U, C256, U, C128, U, C64, C3/C1

Skip layers concatenate output volumes (after activation) of layers 1, 3, 5, 7, 9 of the encoder to the outputs volumes (after activation) of layers 3, 5, 7, 9, 11 of the decoder. We have two separate convolutions as the final layers of the decoder, producing two outputs. The output of C3 is followed by a $tanh$ activation to yield $y_{fg}$. The output of C1 is followed by a $sigmoid$ activation to produce $M_t$.

A.4. Discriminator Network

Input: $[y, p_t] \in \mathbb{R}^{256 \times 256 \times 17}$
Output: $y_{class} \in \mathbb{R}^2$
C64, C128, C256, C256, C256, C256, C256, F, D256, D256

The final dense layer uses a $softmax$ activation.

Appendix B. Additional Results

We present additional results to show that our method is robust to a variety of appearances not shown in the initial results. Fig. 1 shows additional results using our different loss functions. Fig. 2 shows additional cross-action synthesized images.
Figure 1. Additional outputs of our method for different loss functions.
Figure 2. Additional cross-action synthesis outputs of our method.