Supplementary Material

A. More Implementation Details

A.1. SimSaC

SimSaC architecture consists of two decoders: correspondence map decoder (CMD) and mis-correspondence map decoder (MMD). Each estimates the optical flow $\hat{V}$ and the change mask $\hat{M}$, respectively. The decoders exist at each level of the feature pyramid network to implement the coarse-to-fine strategy. In the original paper, the MMD of the first level does not take the estimated flow map from the first-level CMD. This results in that the last-level MMD cannot take advantage of the fine-grained estimate of the optical flow from the last-level CMD. Considering that SimSaC is designed to conduct image alignment based on well-estimated optical flow to perform precise change detection, the inability to utilize the fine-grained optical flow for detecting change would leave the possibility of not using the full potential of the architecture. Therefore, we slightly modify the architecture to utilize the optical flow estimated from the last level of CMD.

At each level $l \geq 1$, CMD^l estimates the flow field $\hat{V}^l$ as follows:

$$\begin{aligned}
\hat{V}^1 &= \text{CMD}^1(c^1), \\
c^1 &= \text{Corr}(F_r^1, F_q^1), \\
\hat{V}^l &= \text{CMD}^l(c^l, \text{up}(\hat{V}^{l-1}), \\
c^l &= \text{Corr}(\tilde{F}_r^l, F_q^l), \\
\tilde{F}_r^l &= \omega(F_r^l, \hat{V}^{l-1}), \text{for } l \geq 2.
\end{aligned} \tag{9}$$

At each level $l \geq 1$, MMD^l preditcs the change mask $\hat{M}^l$ as follows:

$$\begin{aligned}
\hat{M}^l &= \text{MMD}^l(\bar{c}^l, \tilde{F}_r^l, F_q^l), \\
\bar{c}^l &= c^l \odot (\text{up}(M^{l-1})), \\
c^l &= \text{Corr}(\tilde{F}_r^l, F_q^l), \\
\tilde{F}_r^l &= \omega(F_r^l, \hat{V}^l).
\end{aligned} \tag{10}$$

Corr is a correlation layer that computes the dot product of pairs of feature vectors from two feature maps (globally at $l = 1$ and locally for $l \geq 2$), and $\hat{M}^0$ is defined as the mask whose all elements are filled with the value of 1.

A.2. Supervised loss

Given an image pair and the ground truth pixel correspondence map V, a hierarchical end-point error ℓ_V is defined as follows:

$$\ell_V = \sum_{l=1}^{L} \alpha^l \frac{1}{N^l} \sum_{i=1}^{N^l} \|\hat{V}_i^l - V_i^l\|_1, \tag{11}$$

where $\|.\|_1$ is the L1 distance between an estimated flow field $\hat{V}^l$ and the ground truth one V^l, i is an index of pixel location, and N^l is total numbers of pixels at each level l of the L-level feature pyramid. α^l is the weight for each end-point error of pyramid level l.

Given an image pair and the ground-truth pixel-wise change map, a hierarchical focal loss ℓ_M is defined as follows:

$$\begin{aligned}
e_i^l &= (1 - \hat{M_i}^l)^\gamma M_i^l \log(\hat{M_i}^l) \\
&\quad + (\hat{M_i}^l)^\gamma (1 - M_i^l) \log(1 - \hat{M_i}^l), \\
\ell_M &= -\sum_{l=1}^{L} \beta^l \frac{1}{N^l} \sum_{i=1}^{N^l} e_i^l,
\end{aligned} \tag{12}$$

where e_i^l is the focal loss for a pixel i, M_i^l and $\hat{M_i}^{(l)}$ are the ground-truth and estimated change probability of a pixel, respectively, and γ is a constant for reducing the loss for well-classified pixels ($\gamma = 0.5$). Like ℓ_V, we also use the weight β^l for each focal loss of pyramid level l.

A.3. Smoothness Regularization

We utilize the Canny edge detector, which is popular and widely used in computer vision and image processing applications for detecting edges in images. It finds edges by smoothing the image through a Gaussian filter to remove noise and calculating the gradient of the image using Sobel operator. The output of the detector is an edge mask that has values of 0, 0.5, and 1 at edges in images. For fast calculation, we do not use hysteresis thresholding that determines which edges are strong or weak. Since we enforce pixels at edges to have higher gradients, we filter the original edge maps as they only have values of 1 at the outer edges. We implement the Canny edge detector using Kornia, which provides easy access to various computer vision tools.

B. More Qualitative Results

We additionally present qualitative results of the baseline, which are trained by the supervised loss computed by the target labels, and the proposed method that does not utilize any ground truth labels of the target domain. Figures 5, 6, 7, and 8 show the results for ChangeSim-normal, ChangeSim-dusty-air, TSUNAMI, and GSV, respectively.

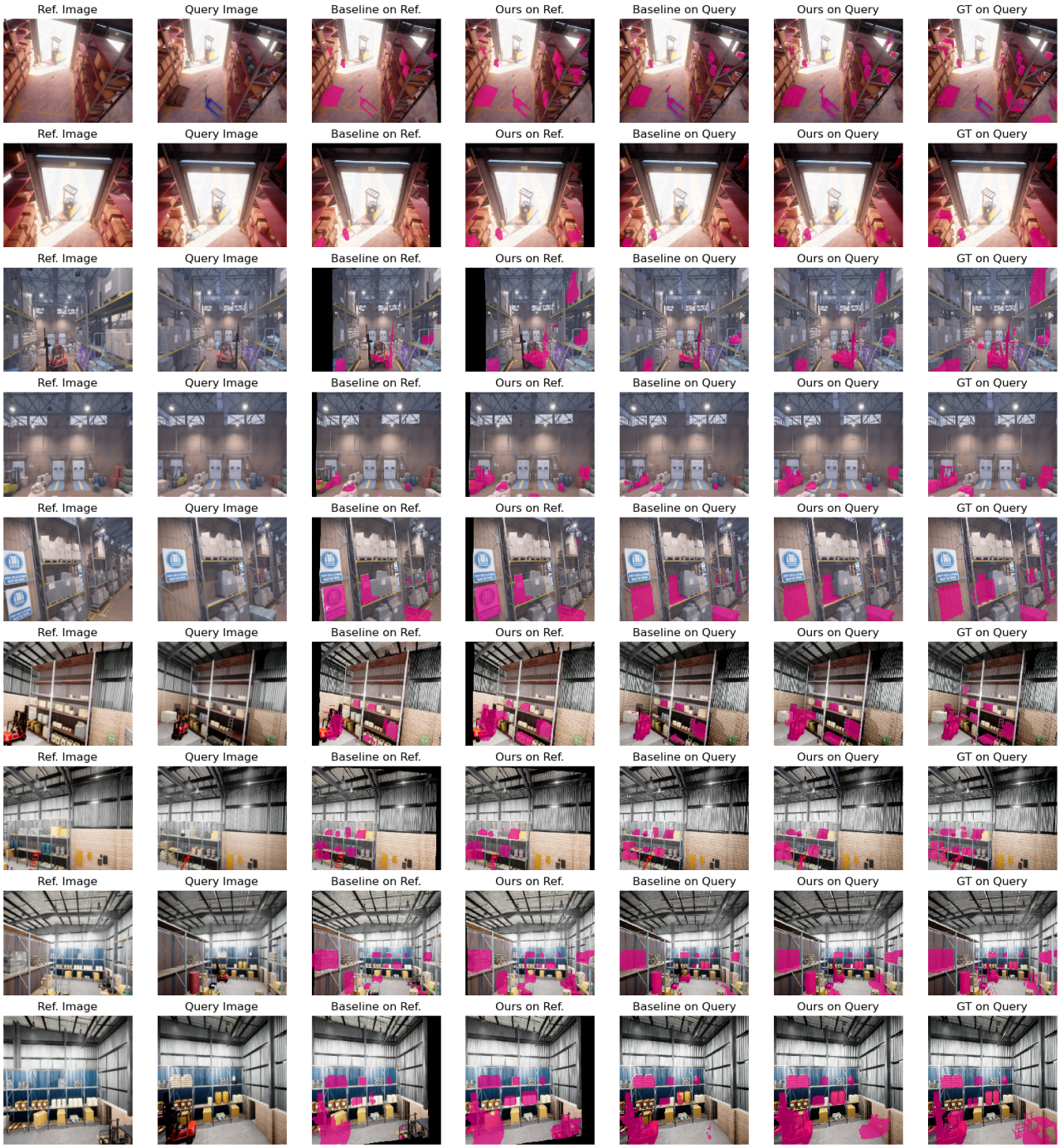

Figure 5. Qualitative results on the ChangeSim-normal dataset.

Figure 6. Qualitative results on the ChangeSim-dusty-air dataset.

Figure 7. Qualitative results on the TSUNAMI dataset.

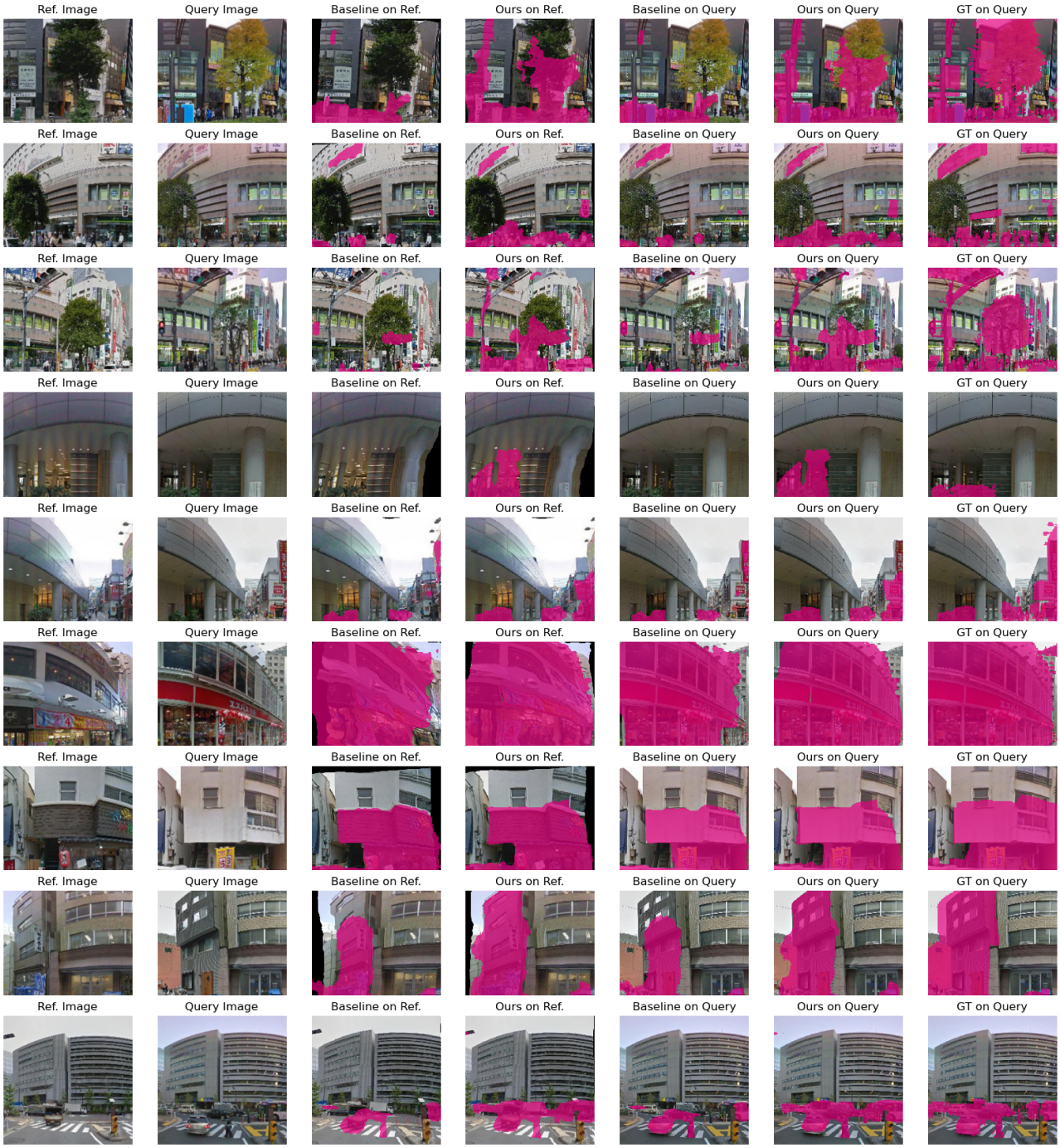

Figure 8. Qualitative results on the GSV dataset.