Supplementary Material
How Shift Equivariance Impacts Metric Learning for Instance Segmentation

Josef Lorenz Rumberger\textsuperscript{1,2}, Xiaoyan Yu\textsuperscript{1,3}, Peter Hirsch\textsuperscript{1,3}, Melanie Dohmen\textsuperscript{1,2}, Vanessa Emanuela Guarino\textsuperscript{1,3}, Ashkan Mokarian\textsuperscript{1}, Lisa Mais\textsuperscript{1}, Jan Funke\textsuperscript{4}, Dagmar Kainmueller\textsuperscript{1}

\textsuperscript{1} Max-Delbrueck-Center for Molecular Medicine in the Helmholtz Association (MDC), Berlin, Germany, \{firstnames.lastname\}@mdc-berlin.de
\textsuperscript{2} Charité University Medicine, Berlin, Germany
\textsuperscript{3} Humboldt-Universität zu Berlin, Faculty of Mathematics and Natural Sciences, Berlin, Germany
\textsuperscript{4} HHMI Janelia Research Campus, Ashburn, VA, USA

1. Examples for equality of U-Net functions \(u\)

\textbf{Example 1:} For a U-Net with identity convolutions, weights 0 for skip connections, and fixed upsampling, functions \(u\) that merely pass the value of a bottleneck pixel through to the output (cf. Fig. 1 in the main paper) are absolute-equal, yet relative-distinct.

\textbf{Example 2a:} For a U-Net with output tile size \(w = 1\) (i.e. a single output pixel per tile), employed in a sliding-window fashion (cf. Sec. 2.1 in the main paper), all functions \(u\) are relative-equal, yet absolute-distinct.

\textbf{Example 2b:} For a U-Net with pooling factor \(f = 1\) (i.e. no pooling and thus full shift equivariance), all functions \(u\) are relative-equal, yet absolute-distinct.

2. Proof of Lemma 1, Part II

An operator \(F\) operating on images \(I\) is \textit{shift equivariant to image shifts} \(t\) iff shifting any input image \(I\) by \(t\) causes an equal or proportional shift \(t'\) in the function’s output, i.e. \(\forall I \forall x : T_{t'}(F(I))(x) = F(T_t(I))(x)\). In case \(t \neq t'\), we call \(F\) shift equivariant to input shifts \(t\) at output shifts \(t'\). In the following, we prove that any U-Net with \(l\) pooling layers and pooling factor \(f\) is shift equivariant to image shifts \(f^l t\), \(t \in \mathbb{Z}\). Without loss of generality, we consider one-dimensional, one-channel input images, and one-channel feature maps throughout. A U-Net is composed of an encoder path and a decoder path:

\textbf{Encoder Path.} An encoder block is composed of a number of conv+ReLU layers. We refer to the function implemented by the \(i\)-th encoder block as \(E_i\). The output of \(E_i\) is passed through a max pooling layer, referred to as \(MP_i\). We refer to the composition of \(E_i\) and \(MP_i\) as \(EMP_i\). The convolution operator \(F_{\text{conv}}\) (stride=1) is commonly defined as \(F_{\text{conv}}(g)(x) = (g \ast h)(x) = \sum_{m \in \mathbb{Z}} g(m) h(m - x)\) (see e.g. [1]), and well-known and easily shown to be shift equivariant to any shifts \(t\):

\[
F_{\text{conv}}(T_t(g))(x) = \sum_{m \in \mathbb{Z}} g(m - t)h(m - x)
\]

\[
= \sum_{m' \in \mathbb{Z}} g(m')h(m' + t - x)
\]

\[
= \sum_{m' \in \mathbb{Z}} g(m')h(m' - (x - t))
\]

\[
= T_t(F_{\text{conv}}(g))(x)
\] (1)

*equal contribution
The same holds for the ReLU operator: $ReLU(T_i(g))(x) = max(0, T_i(g)(x)) = max(0, g(x - t)) = T_i(ReLU(g))(x)$. Compositions of shift equivariant operators are also shift equivariant, hence $E_i$ is shift equivariant to any input shifts $t$. To analyze $MP_t$, we break it down into the max pooling operator $\psi_f(g)(x) := \max\{g(x + i) \mid i \in \{0, \ldots, f - 1\}\}$ with kernel size $f$, and the sub-sampling operator $s_f(g)(x) := g(fx)$ at stride $f$. Analogous to ReLU, $\psi_f$ is shift equivariant to any shift $t$,

$$\psi_f(T_i(g))(x) = \max\{T_i(g)(x + i) \mid i \in \{0, \ldots, f - 1\}\}$$

$$= \max\{g(x + i - t) \mid i \in \{0, \ldots, f - 1\}\}$$

$$= T_i(\psi_f(g))(x),$$

while $T_i(s_f(g))(x) = s_f(T_f((g))(x)$, i.e. $s_f$ is shift equivariant to input shifts $f$ at proportional shifts $t$ of the output. With this we get

$$EMP_i(T_f((g))(x) = s_f(\psi_f(E_i(T_f((g))))(x)$$

$$= s_f(T_f(\psi_f(E_i(T_f((g))))(x)$$

$$= T_i(s_f(\psi_f(E_i((g))))(x)$$

$$= T_i(EMP_i((g))(x),$$

i.e., an encoder block with max pooling with downsampling factor $f$ is shift equivariant to input shifts $f$ at proportional output shifts $t$. Overall, a general encoder path $E$ employs $l$ downsampling operations with factors $f_1, \ldots, f_l$. Shift equivariance proportionality factors multiply when composing operators, hence the encoder path is shift equivariant to input shifts $t \prod_{i=1}^{l} f_i$ at output shifts $t$. In the family of U-Nets we consider, $f_i = f_j$ for all $i, j$, yielding shift equivariance to input shifts $t f^l$ at output shifts $t$: $E(T_f^l(I))(x) = T_l(E(I))(x)$.

**Decoder Path.** The decoder path is composed of decoder blocks $D_i$, whose output is passed through respective upsampling layers, referred to as $UP_i$. A decoder block has the same form as an encoder block, i.e. it consists of a number of conv+ReLU layers, and is thus shift equivariant. We refer to the composition of $D_i$ and $UP_i$ as $DUP_i$. Upsampling is either learnt, i.e. performed via up-convolution with trainable kernel function $p(x)$, with kernel size = stride = $f$ (also called upsampling factor), or performed via nearest neighbor interpolation. We treat both in one go, as the latter is a special case of the former with fixed kernel function $p(x) \equiv 1$. We can express the up-convolution operator $UP_i$ with upsampling factor $f$ as $UP_i((g))(x) = (g * p)(x) = \sum_{m \in \mathbb{Z}} g(m)p(x - fm)$. Concerning its shift equivariance,

$$UP_i(T_i(g))(x) = \sum_{m \in \mathbb{Z}} g(m - t)p(x - fm)$$

$$= \sum_{m' \in \mathbb{Z}} g(m')p(x - f(m' + t))$$

$$= \sum_{m' \in \mathbb{Z}} g(m')p(x - ft - dm')$$

$$= T_{fi}(UP_i(g))(x),$$

i.e., upsampling with factor $f$ is shift equivariant to input shifts $t$ at output shifts $ft$. Thus a decoder block with subsequent upsampling layer, $DUP_i$, is also shift equivariant to input shifts $t$ at output shifts $ft$: $DUP_i(T_f((g))(x) = T_{fi}(DUP_i((g))(x)$. Concerning the input to $DUP_i$, at the bottleneck level $i = l$, this is the output of $EMP_l$. Concerning shift equivariance of their composition $U_i := DUP_i \circ EMP_i$, assuming equal down- and upsampling factors $f$, we get

$$U_i(T_f((g))(x) = DUP_i(EMP_i(T_f((g))(x)$$

$$= DUP_i(T_i(EMP_i((g))(x)$$

$$= T_{fi}(DUP_i(EMP_i((g))(x)$$

$$= T_{fi}(U_i((g))(x),$$

i.e., $U_i$ is shift invariant to shifts $ft$. For $i < l$, the input to $DUP_i$ is a multi-channel image formed by concatenating the output of $DUP_{i+1}$ and the output of encoder block $E_{i+1}$. Referring to the composition of all U-Net blocks up to a block $B$ as $\hat{B}$, we can write the input to $DUP_i$ as $(T_{\Delta_{x_i+1}}(E_i+1(I)), DUP_{i+1}(I))$. Here, $\Delta_{x_i+1}$ is the shift required to centrally align $\hat{E}_{i+1}(I)$ and $DUP_{i+1}(I)$, as $DUP_{i+1}(I)$ is of size smaller or equal than $\hat{E}_{i+1}(I)$. All $\Delta_{x_i}$ are fixed for a given architecture, and hence image concatenation is shift equivariant: $(T_{\Delta_{x_i}}(T_i(g)), T_i(q))(x) = T_i((T_{\Delta_{x_i}}(g), q))(x)$. Consequently, just
like $DUP_i$, for $i < l$, $DUP_i$ is shift equivariant to input shifts $t$ at output shifts $ft$. Furthermore, as proportional shift equivariance factors multiply when composing respective operators, analogous to $U_i := DUP_i \circ EMP_i$, we get that the composition of all blocks from $EMP_i$ to $DUP_i$, $U_i := DUP_i \circ U_{i+1} \circ EMP_i$, is shift equivariant to shifts $f^{l-i+1}t$. In particular, $U_1$ is shift equivariant to shifts $f^1t$.

To yield the full U-Net function $U$, the outputs of $U_1$ and $E_1$ are concatenated into a multi-channel image, which is passed through a final shift invariant decoder block $D_0, U := D_0((\Delta_x, E_1(I), U_1(I)))$ Thus $U$ is equally shift equivariant as $U_1$, i.e., the U-Net is shift equivariant to shifts $f^1t$.

3. Quantitative Evaluation on Benchmark Data

**BBBC006:** The dataset is split into 691 training and 77 test images. Images contain on average 97 instances. We trained a U-Net with $f=2, l=4$, 16-d embeddings and discriminative loss with training tile size 148 and used two-fold cross-validation on the test data to tune hyperparameters.

**DSB2018:** The dataset is split into 380 training, 67 validation and 50 test images. Images contain on average 49 instances. We trained a U-Net with $f=2, l=4$, 16-d embeddings and discriminative loss with training tile size 68. Aside from the loss the same setup as in [3] is used.

**nuclei3d:** The dataset is split into 18 training, 3 validation and 7 test volumes. Volumes contain on average 537 instances. We trained a U-Net with $f=2, l=3$, 16-d embeddings and discriminative loss with training tile size 148. Aside from the loss the same setup as in [2] is used.

Figure 1: Predicted embeddings of a U-Net (a) naively stitched without cropping and (b) cropped to $n \cdot f^l$ before stitching. Inconsistencies at the stitching boundaries are clearly visible in the (c) gradient magnitudes of the embeddings.

4. Zero padding leads to location awareness

Figure 2: A U-Net with zero-padding yields location awareness also with nearest-neighbor upsampling, if (a) each output pixel has a unique receptive field that reaches the image boundary. Otherwise, for a constant input image, (b) some pixels will necessarily receive equal outputs. Showcase: $l = 2, f = 2$, input image $I \equiv 1$. 
5. Practical Impact of Noise and Small Deformations

![Image](image.png)

(a) Slight gaussian noise added to the input: Instances distinguished despite object spacing $f^l$

(b) Elastic deformations: Instances distinguished despite object spacing $f^l$

(c) Slight gaussian noise: No effect on stitching issues

(d) Elastic deformations on the input: No effect on stitching issues

Figure 3: (a) and (b): Slight image augmentations enable a U-Net to distinguish objects that are otherwise indistinguishable due to object spacing $f^l$. Showcase: $l = 4$, $f = 2$, learnt upsampling. (a) Slight Gaussian noise, and (b) elastic deformations, best viewed on screen with zoom. (c) and (d): However, image augmentations do not affect the issue of inconsistencies in a tile-and-stitch approach if output tiles are not cropped to edge length $n \cdot f^l$ before stitching. Showcase: $l = 4$, $f = 2$, inference output tile size $52 \neq n \cdot 2^4$.

References

