

Supplementary material

Deep learning based domain adaptation for mitochondria segmentation on EM volumes

Daniel Franco-Barranco^{1,2}, Julio Pastor-Tronch¹, Aitor Gonzalez-Marfil¹,
Arrate Muñoz-Barrutia^{3,4}, and Ignacio Arganda-Carreras^{1,2,5}

¹ Dept. of Computer Science and Artificial Intelligence, University of the Basque Country (UPV/EHU)

² Donostia International Physics Center (DIPC)

³ Universidad Carlos III de Madrid

⁴ Instituto de Investigación Sanitaria Gregorio Marañón

⁵ Ikerbasque, Basque Foundation for Science

daniel_franco001@ehu.eus

S1 Cross-dataset results

To complete the overview of results shown in the manuscript, in this section, we show examples of histogram-matching and style-transfer image transformations, together with segmentation results of all cross-dataset experiments. Full-size images are shown for qualitative evaluation purposes.

S1.1 Source: Lucchi++ - Target: Kasthuri++

The effect of our histogram-matching and style-transfer methods on an image from the Kasthuri++ dataset is shown in Figure S1.1 using the Lucchi++ dataset as the source domain. Remarkably, the domain shift in this source-target combination seems to be the smallest of all cases, and the histogram-matched images (see Figures S1.1a, S1.1b) appear to be very close to the source domain images.

The mitochondria probability maps produced by all our tested methods on the first test image from Kasthuri++ are shown in Figure S1.2 together with its corresponding ground-truth binary labels and original EM image. The best qualitative results seem to be produced by the histogram-matching and style-transfer approaches (see Figures S1.2b, S1.2c), while the state-of-the-art DAMT-Net method struggles to produce compact mitochondria masks and presents border artifacts due to the zero-padding of the Kasthuri++ dataset (see Figure S1.2f). Notice that the displayed results for the style-transfer, SSL, Attention Y-Net, and DAMT-Net approaches correspond to executions using our proposed stop criterion (solidity, see Section 4.3).

S1.2 Source: Lucchi++ - Target: VNC

The effect of our histogram-matching and style-transfer methods on an image from the VNC dataset is shown in Figure S1.3 using the Lucchi++ dataset as the source domain. The domain shift in this source-target combination seems much larger than in the previous case, and the histogram-matched images (see Figures S1.3a, S1.3c) appear to be further away from the source domain images than the stylized ones (see Figure S1.3d). In particular, the style-transfer method successfully adapted the texture of the neural process from one domain (ssTEM) to the other (FIB-SEM).

The mitochondria probability maps produced by all our tested methods on the first test image from VNC are shown in Figure S1.4 together with its corresponding ground-truth binary labels and original EM image. Although all methods seem to approximate the location of mitochondria correctly, the best qualitative results seem to be produced by our style-transfer approach (see Figure S1.4c). In this case, the state-of-the-art DAMT-Net method seems to produce under-segmented results (see Figure S1.4f) while our SSL and Attention Y-Net (see Figures S1.4d, S1.4e) methods output over-segmented masks. Notice that the displayed results for the style-transfer, SSL, Attention Y-Net, and DAMT-Net approaches correspond to executions using our proposed stop criterion (solidity, see Section 4.3).

S1.3 Source: Kasthuri++ - Target: Lucchi++

The effect of our histogram-matching and style-transfer methods on an image from the Lucchi++ dataset is shown in Figure S1.5 using the Kasthuri++ dataset as the source domain. The domain shift in this source-target combination seems larger than in the opposite combination, where the histogram matching obtained excellent results. As in the previous case, here, the histogram-matched images (see Figures S1.5a, S1.5c) appear to be far away from the source domain images. However, the style-transfer method (see Figure S1.5d) has managed to successfully capture the texture of both the neural process and organelles from one domain (FIB-SEM) to the other (ssEM).

The mitochondria probability maps produced by all our tested methods on the first test image from Lucchi++ are shown in Figure S1.6 together with its corresponding ground-truth binary labels and original EM image. In these experiments, all learning-based methods perform notably well (see Figures S1.6c-S1.6f). Nevertheless, the best qualitative results appear to be those produced by our Attention Y-Net approach (see Figure S1.6e), which are very close to the desired ground truth output (Figure S1.6g). Notice that the displayed results for the style-transfer, SSL, Attention Y-Net, and DAMT-Net approaches correspond to executions using our proposed stop criterion (solidity, see Section 4.3).

S1.4 Source: Kasthuri++ - Target: VNC

The effect of our histogram-matching and style-transfer methods on an image from the VNC dataset is shown in Figure S1.7 using the Kasthuri++ dataset

as the source domain. The domain shift seems quite large in this case, and the histogram-matched images (Figure S1.7c) appear to be far away from the source domain images (Figure S1.7a). In appearance, the style-transfer results (Figure S1.7d) do not look much better either, but the results from Table 1 indicate the method was quite successful at transferring the style from the ssEM to the ssTEM dataset.

The mitochondria probability maps produced by all our tested methods on the first test image from VNC are shown in Figure S1.8 together with its corresponding ground-truth binary labels and original EM image. In these experiments, most methods struggle to produce proper mitochondria masks. The exception is our style-transfer approach (Figure S1.8c), which correctly finds all mitochondria present in the ground truth (Figure S1.8g) but also produces a couple of large mitochondria-like artifacts. Notice that the displayed results for the style-transfer, SSL, Attention Y-Net, and DAMT-Net approaches correspond to executions using our proposed stop criterion (solidity, see Section 4.3).

S1.5 Source: VNC - Target: Lucchi++

The effect of our histogram-matching and style-transfer methods on an image from the Lucchi++ dataset is shown in Figure S1.9 using the VNC dataset as the source domain. Both the histogram-matched image (Figure S1.9c) and the stylized image (Figure S1.9d) seem to reproduce the appearance of the source domain image (Figure S1.9a). In particular, the style-transfer results (Figure S1.9d) are able to not only reproduce the source intensities but also correctly replicate the textures inside the neural processes.

The mitochondria probability maps produced by all our tested methods on the first test image from Lucchi++ are shown in Figure S1.10 together with its corresponding ground-truth binary labels and original EM image. While all methods identify all the mitochondria present in the ground truth correctly (Figure S1.10g), most of them produce an over-segmentation, except for DAMT-Net, which is under-segmenting (Figure S1.10f). Although some extra low-probability maps are created by the SSL method (Figure S1.10d), its medium-high probability maps nicely capture the real mitochondria. Notice that the displayed results for the style-transfer, SSL, Attention Y-Net, and DAMT-Net approaches correspond to executions using our proposed stop criterion (solidity, see Section 4.3).

S1.6 Source: VNC - Target: Kasthuri++

The effect of our histogram-matching and style-transfer methods on an image from the Kasthuri++ dataset is shown in Figure S1.11 using the VNC dataset as the source domain. As in the previous case, both the histogram-matched image (Figure S1.11c) and the stylized image (Figure S1.11d) seem to reproduce the appearance of the source domain image (Figure S1.11a). Again, the style-transfer results (Figure S1.11d) seem to not only reproduce the source intensities but also correctly replicate the textures inside the neural processes.

The mitochondria probability maps produced by all our tested methods on the first test image from Kasthuri++ are shown in Figure S1.12 together with its corresponding ground-truth binary labels and original EM image. Here all methods struggle to correctly identify the mitochondria present in the ground truth (Figure S1.12g). Some of them produce an over-segmentation (Figures S1.12a, S1.12c, S1.12e), while others are under-segmenting (Figures S1.12d, S1.12f). As observed before, the DAMT-Net method produces artifacts in the border of the tissue areas due to the padding (Figure S1.12f). Notice that the displayed results for the style-transfer, SSL, Attention Y-Net, and DAMT-Net approaches correspond to executions using our proposed stop criterion (solidity, see Section 4.3).

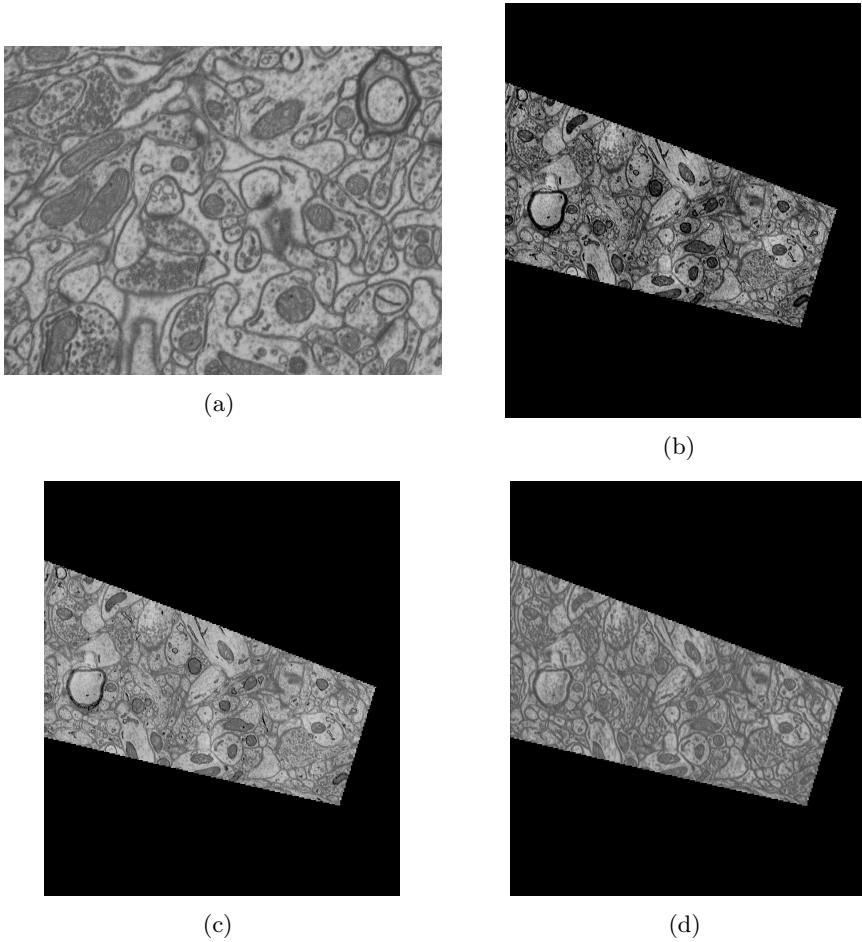


Fig. S1.1: Examples of histogram-matching and style-transfer results using Lucchi++ as reference histogram/style to transform Kasthuri++ images: (a) Lucchi++ dataset sample; (b) original Kasthuri++ test image; (c) histogram-matched version of (b); and (d) stylized version of (b).

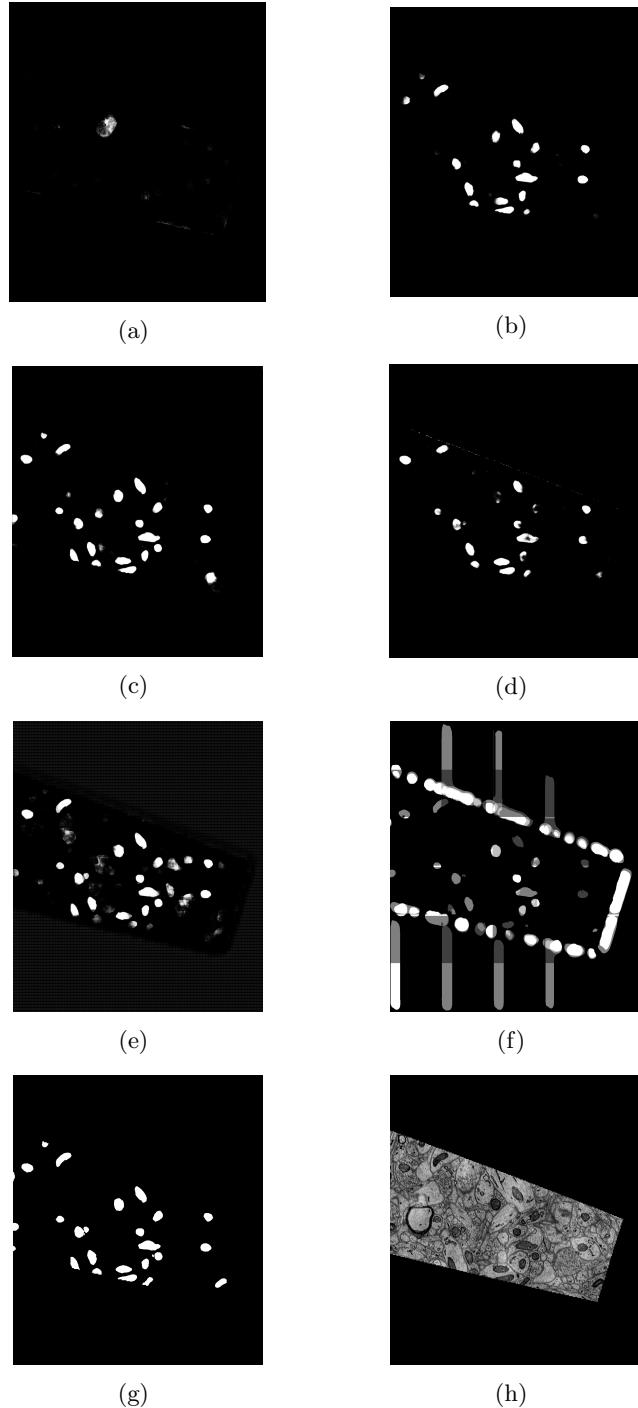


Fig. S1.2: Examples of semantic segmentation results using Lucchi++ as source and Kasthuri++ as the target. The resulting mitochondria probability maps are shown for: (a) the baseline method (no adaptation); (b) the baseline method applied to the histogram-matched images; our (c) style-transfer, (d) self-supervised learning, and (e) Attention Y-Net approaches; and (f) the DAMT-Net method; together with the corresponding (g) ground truth and (h) original test sample from Kasthuri++.

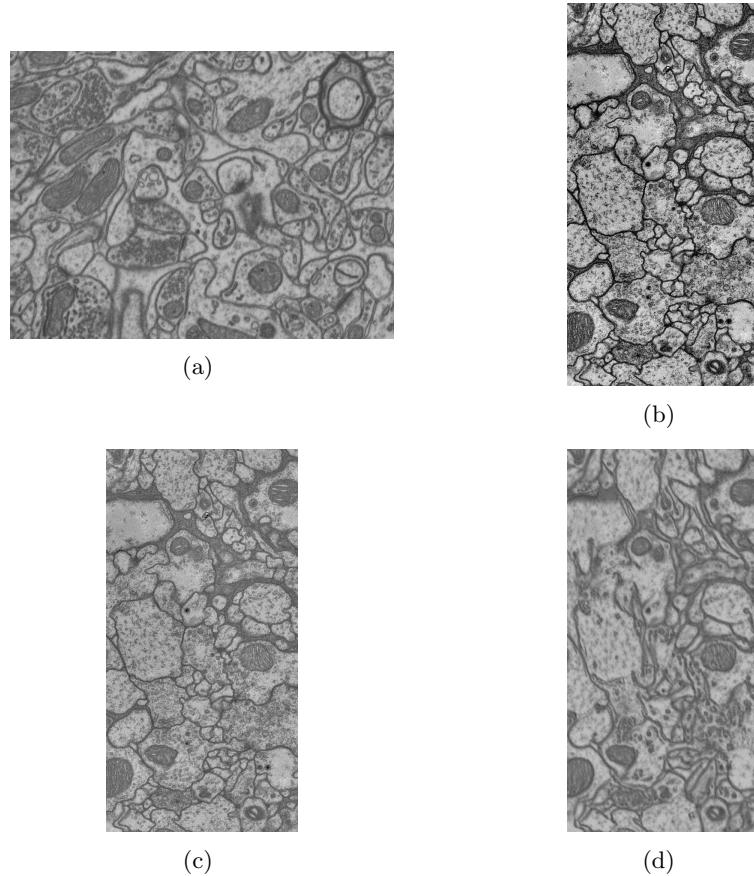


Fig. S1.3: Examples of histogram-matching and style-transfer results using Lucchi++ as reference histogram/style to transform VNC images: (a) Lucchi++ dataset sample; (b) original VNC test image; (c) histogram-matched version of (b); and (d) stylized version of (b).

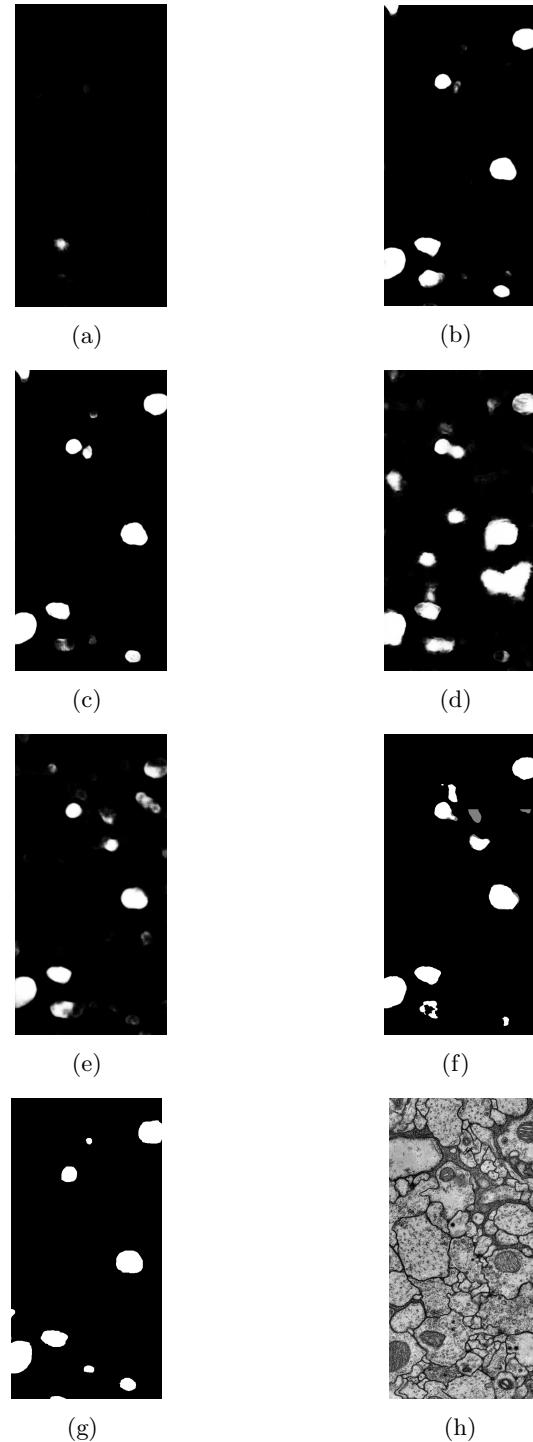


Fig. S1.4: Examples of semantic segmentation results using Lucchi++ as the source and VNC as the target. The resulting mitochondria probability maps are shown for: (a) the baseline method (no adaptation); (b) the baseline method applied to the histogram-matched images; our (c) style-transfer, (d) self-supervised learning, and (e) Attention Y-Net approaches; and (f) the DAMT-Net method; together with the corresponding (g) ground truth and (h) original test sample from VNC.

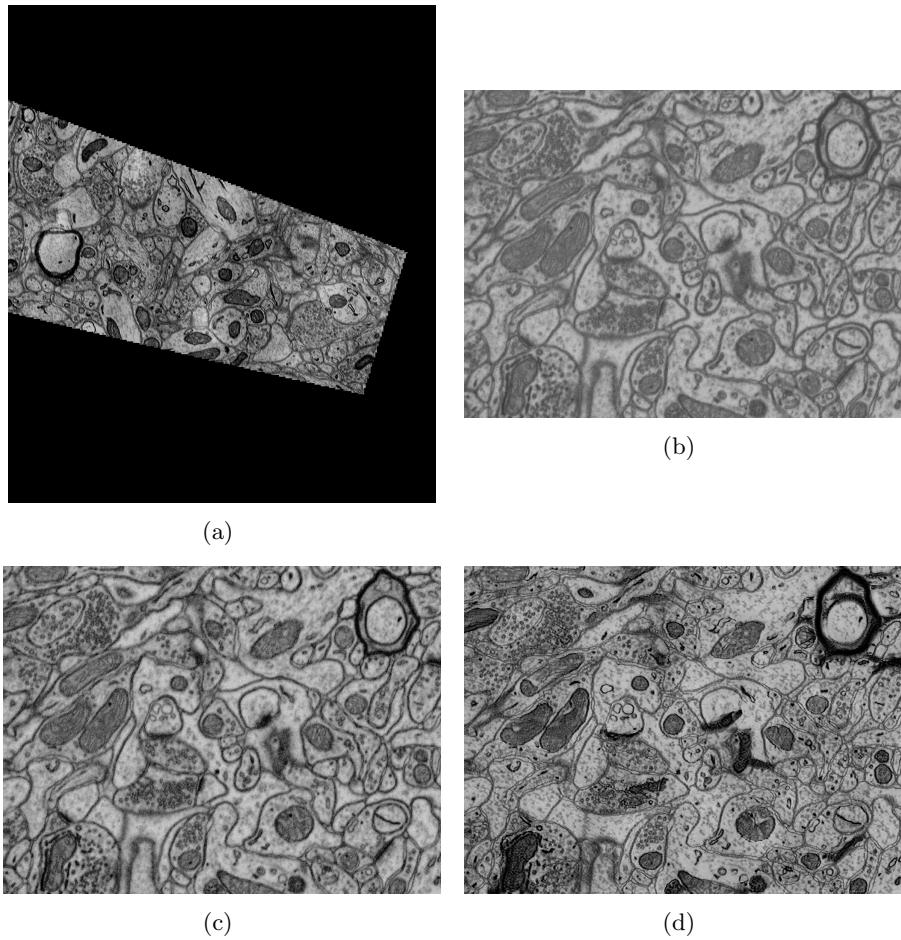


Fig. S1.5: Examples of histogram-matching and style-transfer results using Kasthuri++ as reference histogram/style to transform Lucchi++ images: (a) Kasthuri++ dataset sample; (b) original Lucchi++ test image; (c) histogram-matched version of (b); and (d) stylized version of (b).

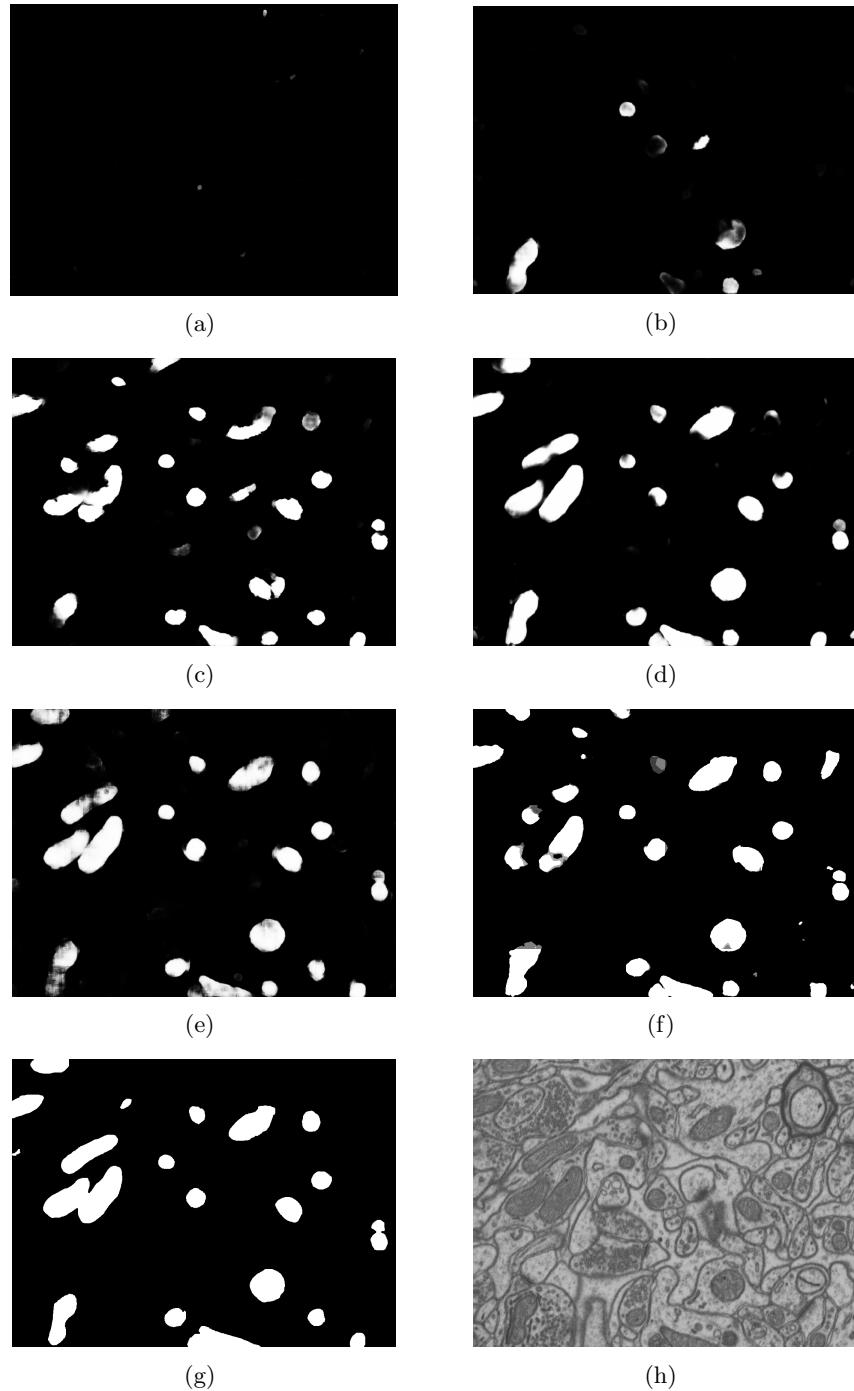


Fig. S1.6: Examples of semantic segmentation results using Kasthuri++ as the source and Lucchi++ as the target. The resulting mitochondria probability maps are shown for: (a) the baseline method (no adaptation); (b) the baseline method applied to the histogram-matched images; our (c) style-transfer, (d) self-supervised learning, and (e) Attention Y-Net approaches; and (f) the DAMT-Net method; together with the corresponding (g) ground truth and (h) original test sample from Lucchi++.

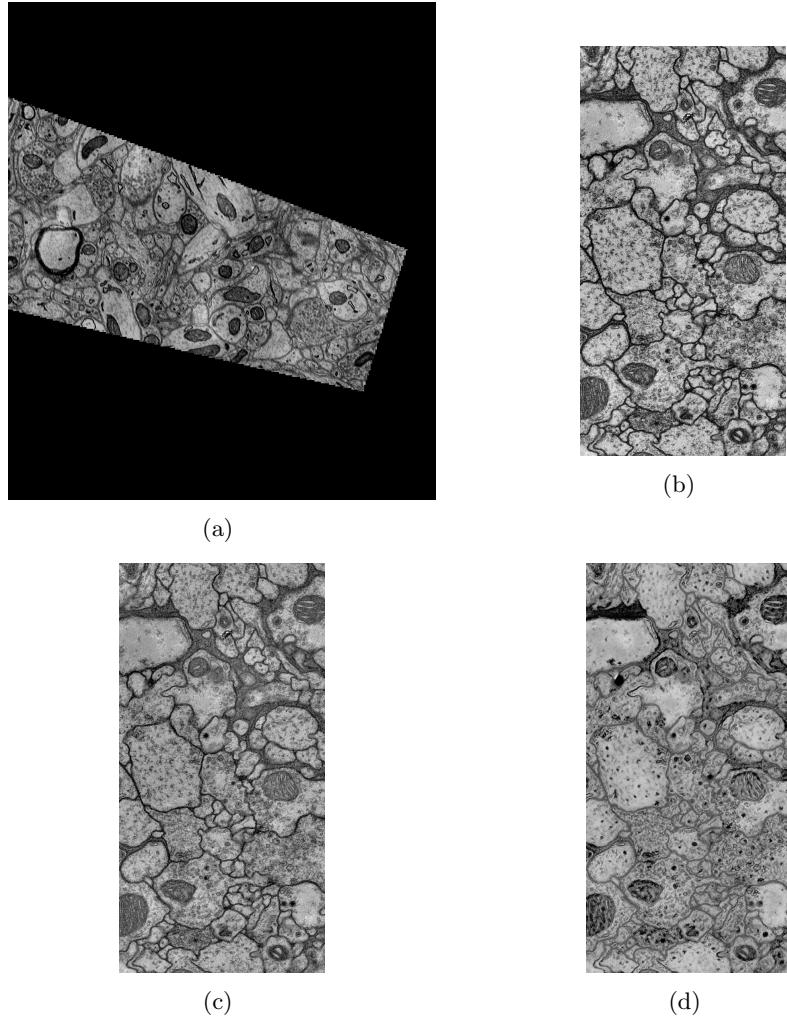


Fig. S1.7: Examples of histogram-matching and style-transfer results using Kasthuri++ as reference histogram/style to transform VNC images: (a) Kasthuri++ dataset sample; (b) original VNC test image; (c) histogram-matched version of (b); and (d) stylized version of (b).

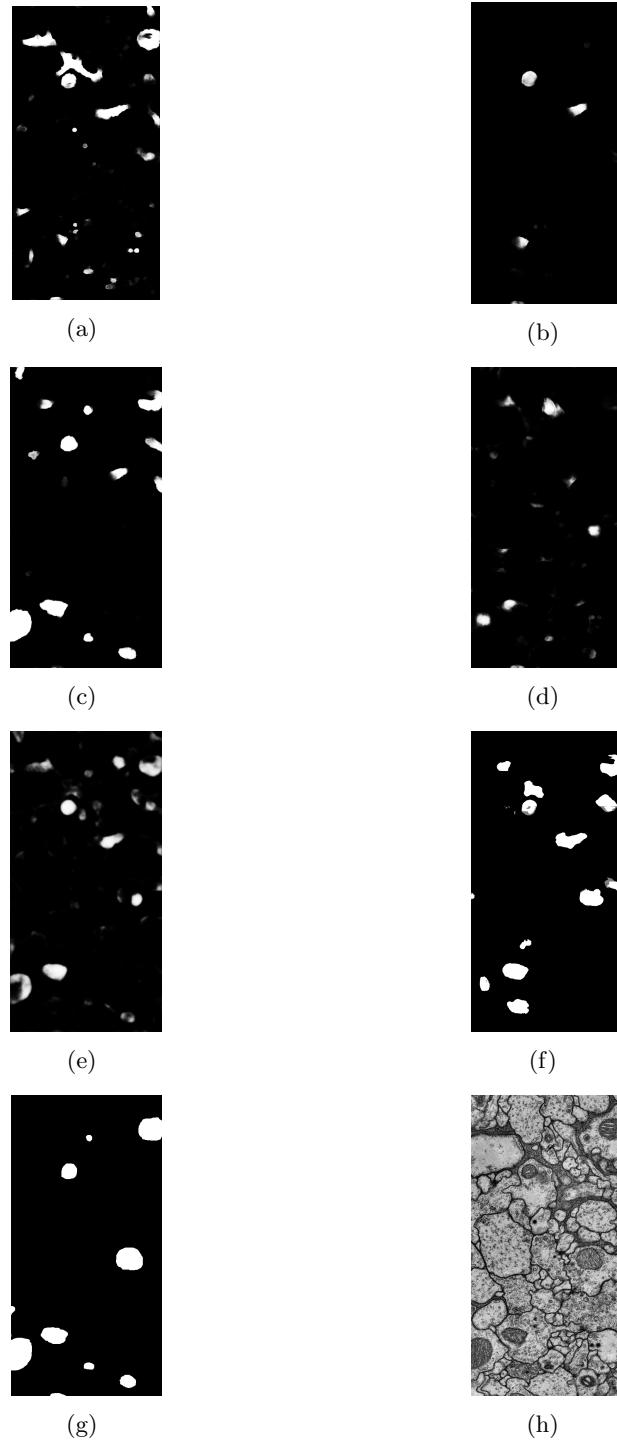


Fig. S1.8: Examples of semantic segmentation results using Kasthuri++ as the source and VNC as the target. The resulting mitochondria probability maps are shown for: (a) the baseline method (no adaptation); (b) the baseline method applied to the histogram-matched images; our (c) style-transfer, (d) self-supervised learning, and (e) Attention Y-Net approaches; and (f) the DAMT-Net method; together with the corresponding (g) ground truth and (h) original test sample from VNC.

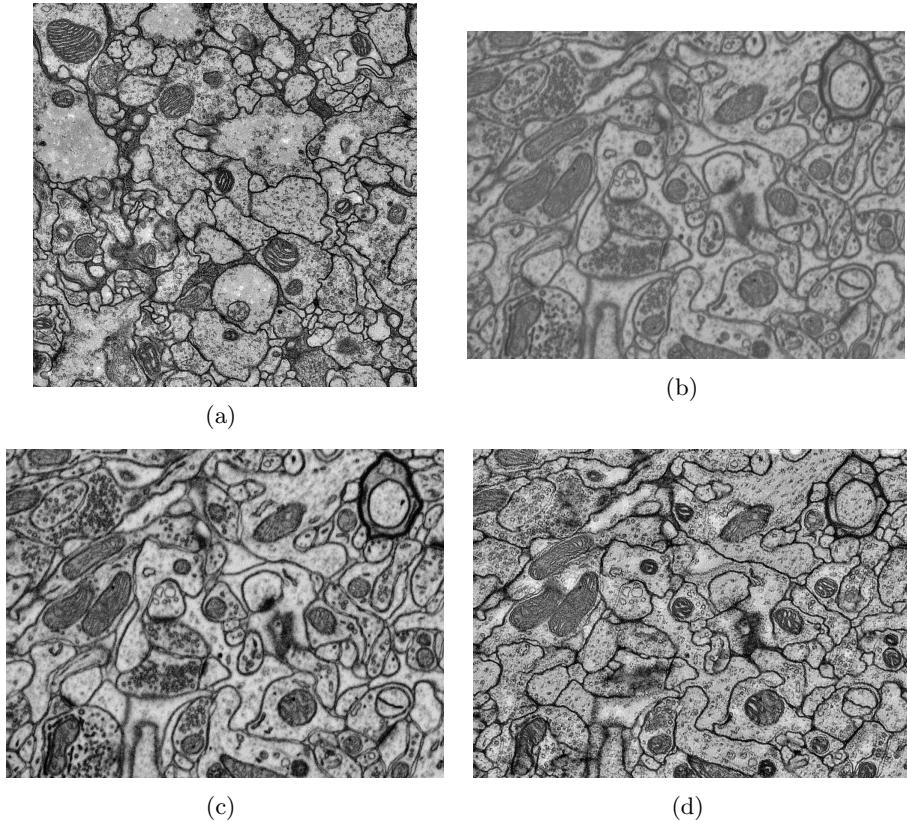


Fig. S1.9: Examples of histogram-matching and style-transfer results using VNC as reference histogram/style to transform Lucchi++ images. From top to bottom and from left to right: (a) VNC dataset sample; (b) original Lucchi++ test image; (c) histogram-matched version of (b); and (d) stylized version of (b).

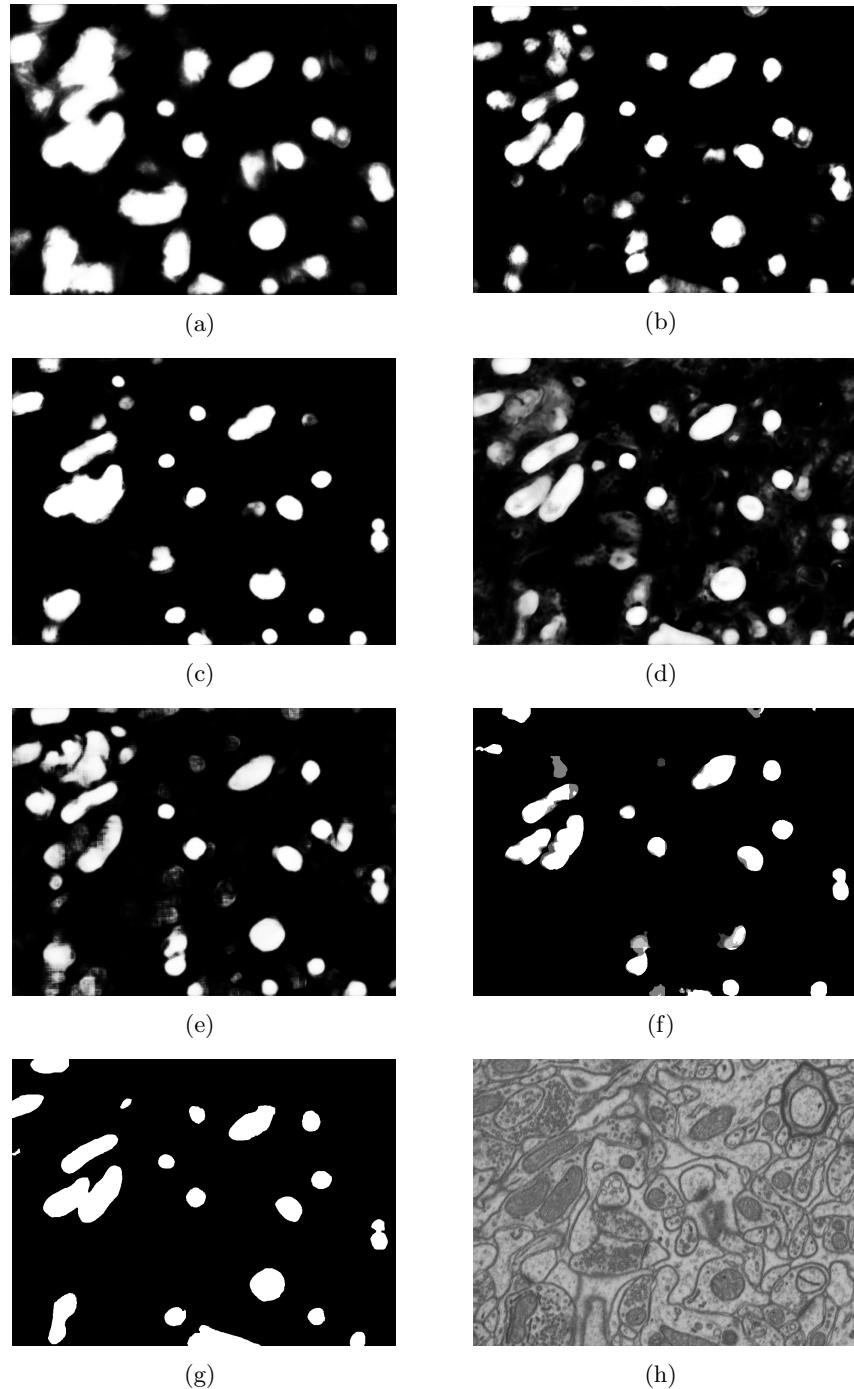


Fig. S1.10: Examples of semantic segmentation results using VNC as the source and Lucchi++ as the target. The resulting mitochondria probability maps are shown for: (a) the baseline method (no adaptation); (b) the baseline method applied to the histogram-matched images; our (c) style-transfer, (d) self-supervised learning, and (e) Attention Y-Net approaches; and (f) the DAMT-Net method; together with the corresponding (g) ground truth and (h) original test sample from Lucchi++.

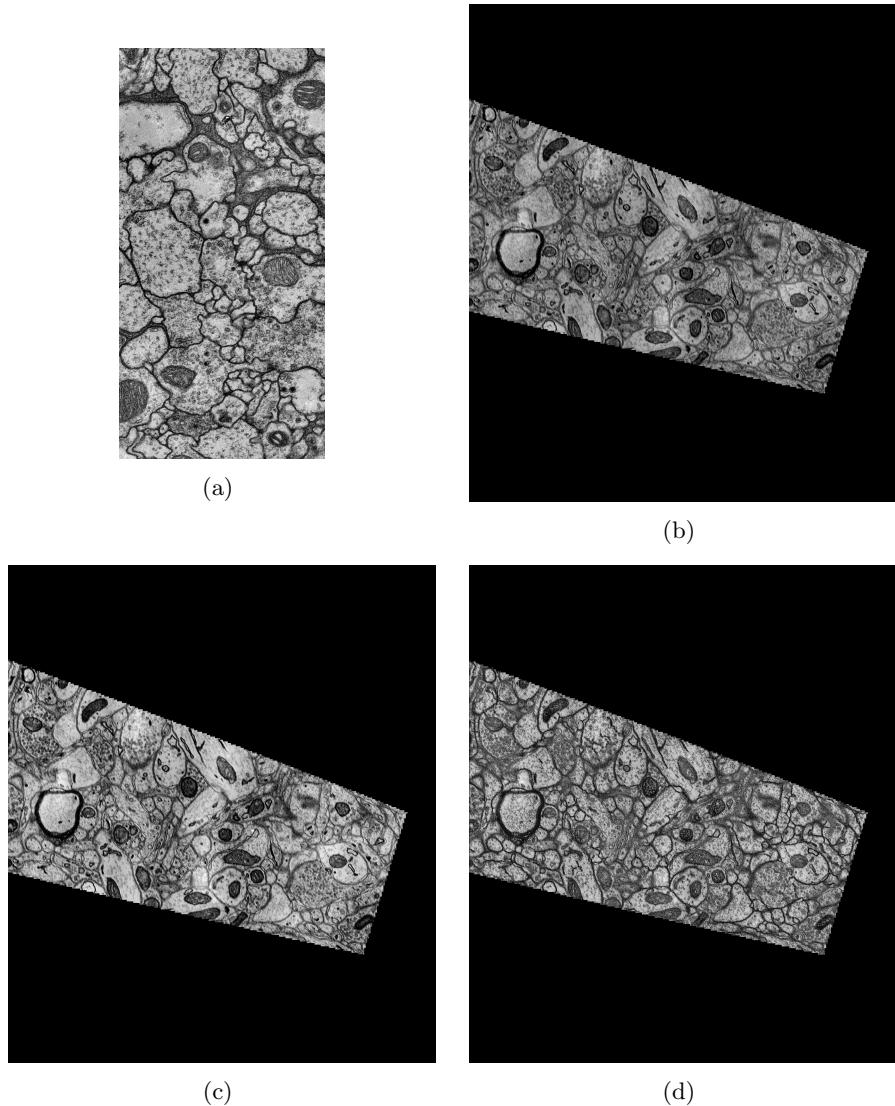


Fig. S1.11: Examples of histogram-matching and style-transfer results using VNC as reference histogram/style to transform Kasthuri++ images. From top to bottom and from left to right: (a) VNC dataset sample; (b) original Kasthuri++ test image; (c) histogram-matched version of (b); and (d) stylized version of (b).

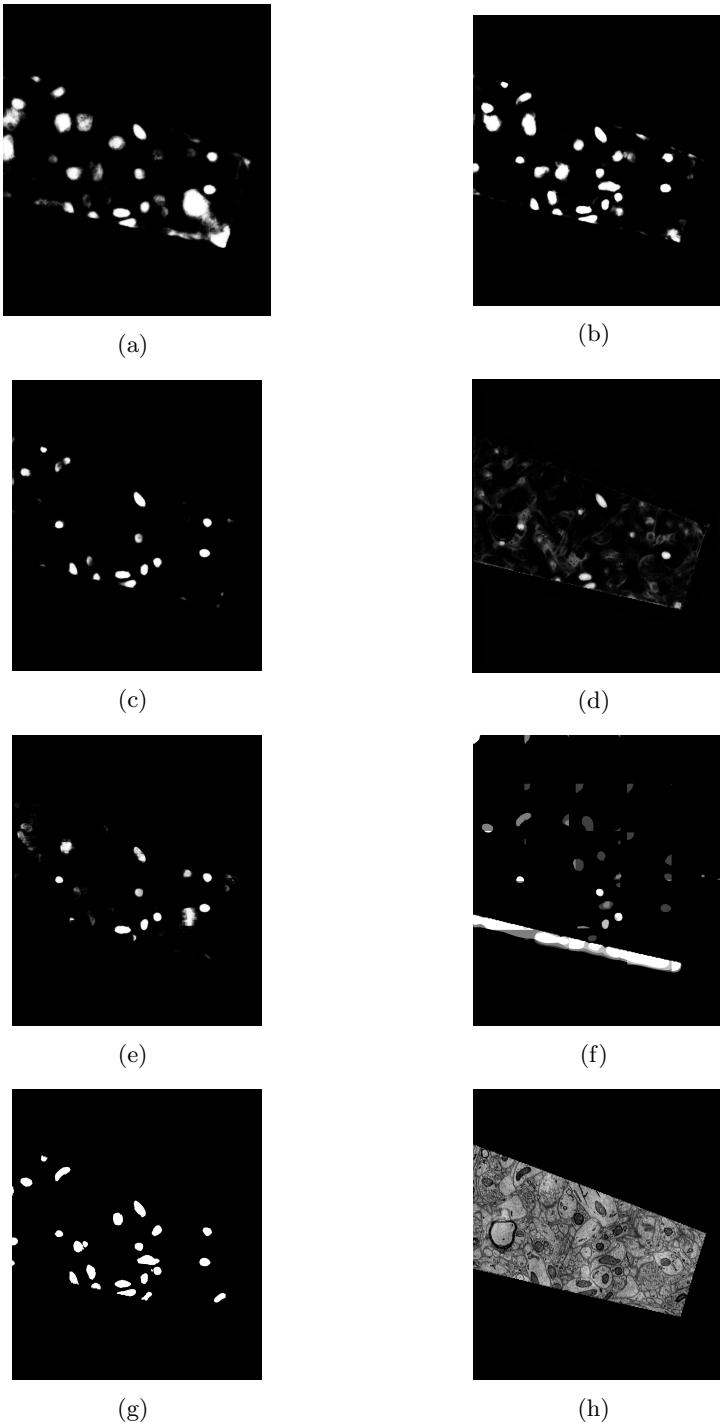


Fig. S1.12: Examples of semantic segmentation results using VNC as the source and Kasthuri++ as the target. The resulting mitochondria probability maps are shown for: (a) the baseline method (no adaptation); (b) the baseline method applied to the histogram-matched images; our (c) style-transfer, (d) self-supervised learning, and (e) Attention Y-Net approaches; and (f) the DAMT-Net method; together with the corresponding (g) ground truth and (h) original test sample from Kasthuri++.

S2 Analysis of solidity as stop condition

In this section, we analyze the effect of using the solidity of the predicted masks (see Section 4.3) as a stop condition in all the tested learning methods. With that aim, we plot the solidity values at each epoch of every cross-dataset experiment and, on a complementary plot, the IoU values produced in the test partition of the target dataset at the same epochs.

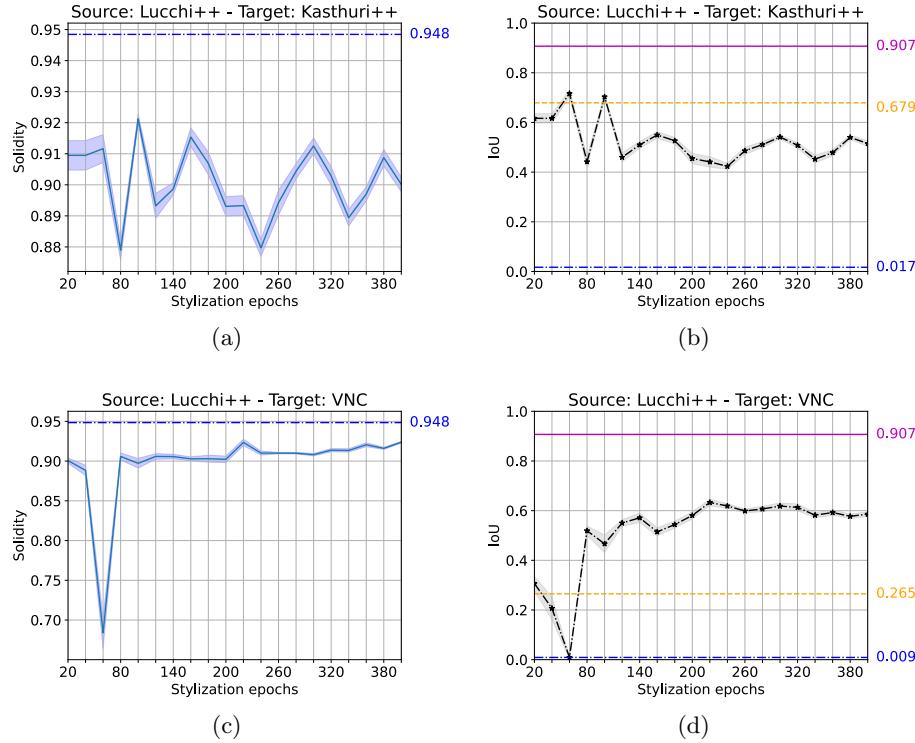


Fig. S2.1: Relation between solidity and IoU in the style-transfer approach with Lucchi++ as source domain. On the left, the evolution of the solidity value (averaged for ten executions) as a function of the stylization epochs with (a) Kasthuri++ and (c) VNC as target domains (dashed lines represent the source solidity value). On the right, the evolution of the test IoU (also averaged over ten executions) as a function of the epochs with (b) Kasthuri++ and (d) VNC as target domains. The magenta lines represent the maximum IoU value obtained by the fully supervised baseline models. In contrast, the blue and orange lines represent the IoU values obtained by the baseline methods applied without adaptation and after histogram matching to the target datasets, respectively.

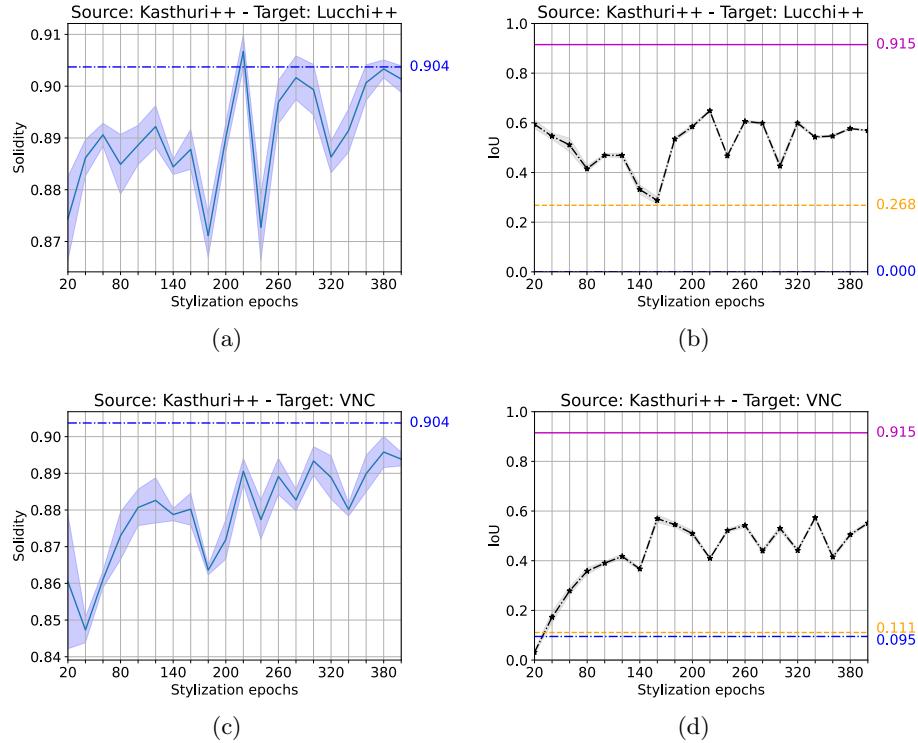


Fig. S2.2: Relation between solidity and IoU in the style-transfer approach with Kasthuri++ as source domain. On the left, the evolution of the solidity value (averaged for ten executions) as a function of the stylization epochs with (a) Lucchi++ and (c) VNC as target domains (dashed lines represent the source solidity value). On the right, the evolution of the test IoU (also averaged over ten executions) as a function of the epochs with (b) Lucchi++ and (d) VNC as target domains. The magenta lines represent the maximum IoU value obtained by the fully supervised baseline models. In contrast, the blue and orange lines represent the IoU values obtained by the baseline methods applied without adaptation and after histogram matching to the target datasets, respectively.

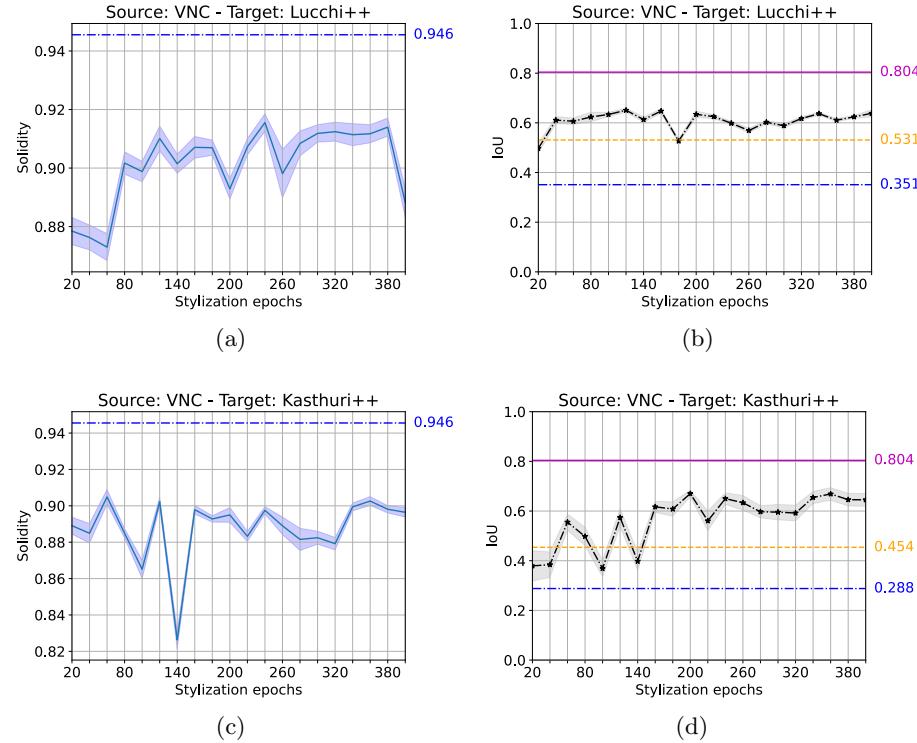


Fig. S2.3: Relation between solidity and IoU in the style-transfer approach with VNC as source domain. On the left, the evolution of the solidity value (averaged for ten executions) as a function of the stylization epochs with (a) Lucchi++ and (c) Kasthuri++ as target domains (dashed lines represent the source solidity value). On the right, the test IoU evolution (averaged over ten executions) as a function of the epochs with (b) Lucchi++ and (d) Kasthuri++ as target domains. The magenta lines represent the maximum IoU value obtained by the fully supervised baseline models. In contrast, the blue and orange lines represent the IoU values obtained by the baseline methods applied without adaptation and after histogram matching to the target datasets, respectively.

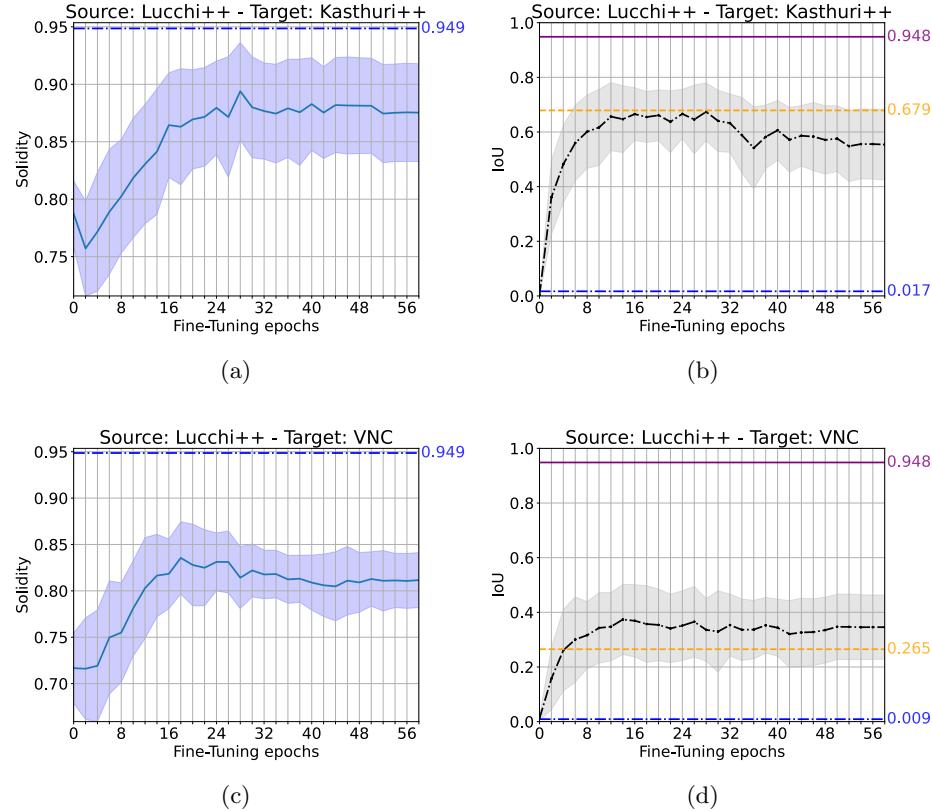


Fig. S2.4: Relation between solidity and IoU in the self-supervised approach with Lucchi++ as source domain. On the left, the evolution of the solidity value (averaged for ten executions) as a function of the stylization epochs with (a) Kasthuri++ and (c) VNC as target domains (dashed lines represent the source solidity value). On the right, the evolution of the test IoU (also averaged over ten executions) as a function of the epochs with (b) Kasthuri++ and (d) VNC as target domains. The magenta lines represent the maximum IoU value obtained by the fully supervised baseline models. In contrast, the blue and orange lines represent the IoU values obtained by the baseline methods applied without adaptation and after histogram matching to the target datasets, respectively.

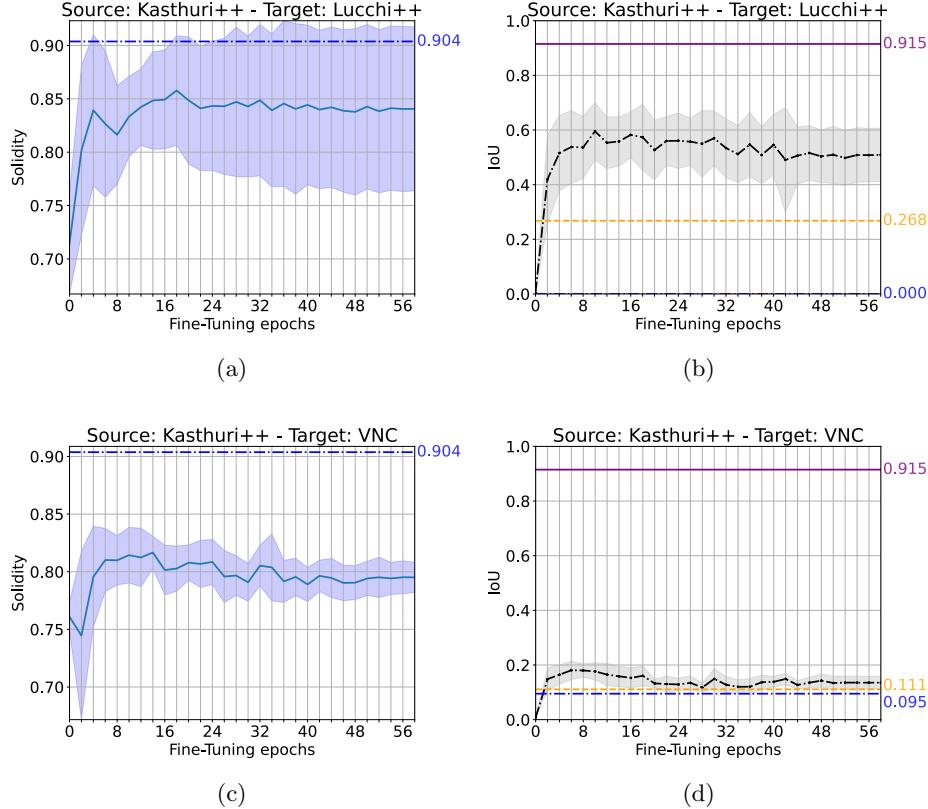


Fig. S2.5: Relation between solidity and IoU in the self-supervised approach with Kasthuri++ as source domain. On the left, the evolution of the solidity value (averaged for ten executions) as a function of the stylization epochs with (a) Kasthuri++ and (c) VNC as target domains (dashed lines represent the source solidity value). On the right, the evolution of the test IoU (also averaged over ten executions) as a function of the epochs with (b) Kasthuri++ and (d) VNC as target domains. The magenta lines represent the maximum IoU value obtained by the fully supervised baseline models. In contrast, the blue and orange lines represent the IoU values obtained by the baseline methods applied without adaptation and after histogram matching to the target datasets, respectively.

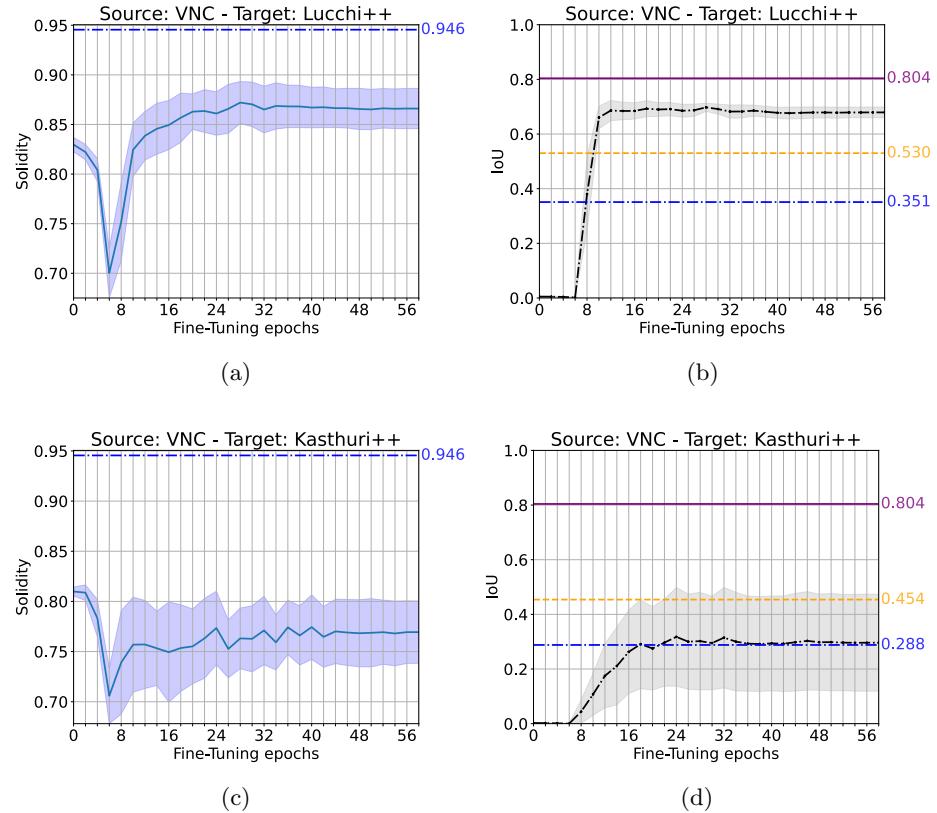


Fig. S2.6: Relation between solidity and IoU in the self-supervised approach with VNC as source domain. On the left, the evolution of the solidity value (averaged for ten executions) as a function of the stylization epochs with (a) Lucchi++ and (c) Kasthuri++ as target domains (dashed lines represent the source solidity value). On the right, the test IoU evolution (averaged over ten executions) as a function of the epochs with (b) Lucchi++ and (d) Kasthuri++ as target domains. The magenta lines represent the maximum IoU value obtained by the fully supervised baseline models. In contrast, the blue and orange lines represent the IoU values obtained by the baseline methods applied without adaptation and after histogram matching to the target datasets, respectively.

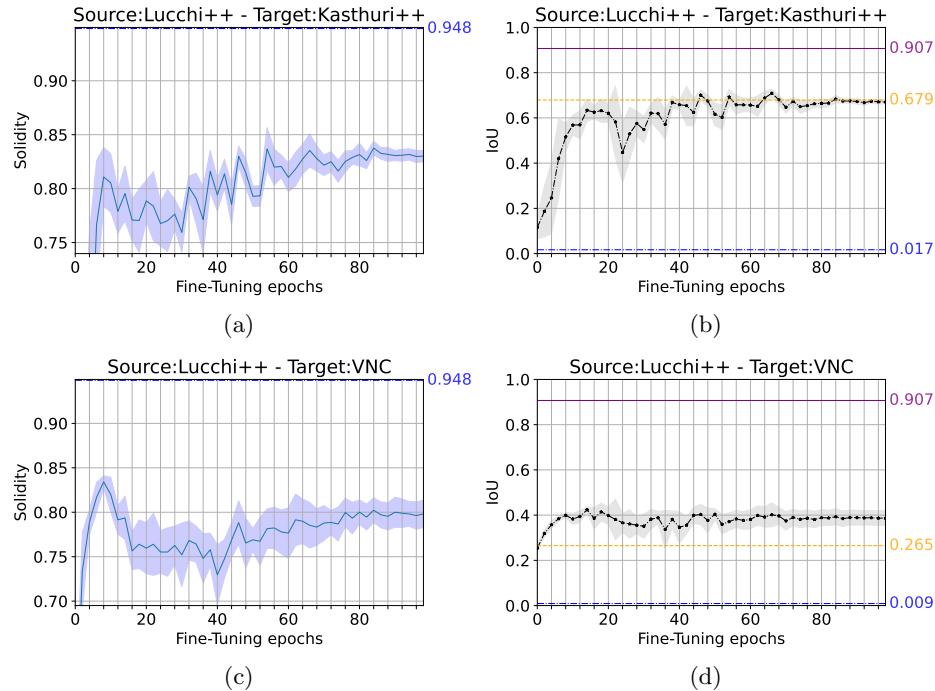


Fig. S2.7: Relation between Solidity and IoU in the Attention Y-Net approach with Lucchi++ as source domain. On the left, the evolution of the Solidity value (averaged for ten executions) as a function of the stylization epochs with (a) Kasthuri++ and (c) VNC as target domains (dashed lines represent the source Solidity value). On the right, the evolution of the test IoU (also averaged over ten executions) as a function of the epochs with (b) Kasthuri++ and (d) VNC as target domains. The magenta lines represent the maximum IoU value obtained by the fully supervised baseline models. In contrast, the blue and orange lines represent the IoU values obtained by the baseline methods applied without adaptation and after histogram matching to the target datasets, respectively.

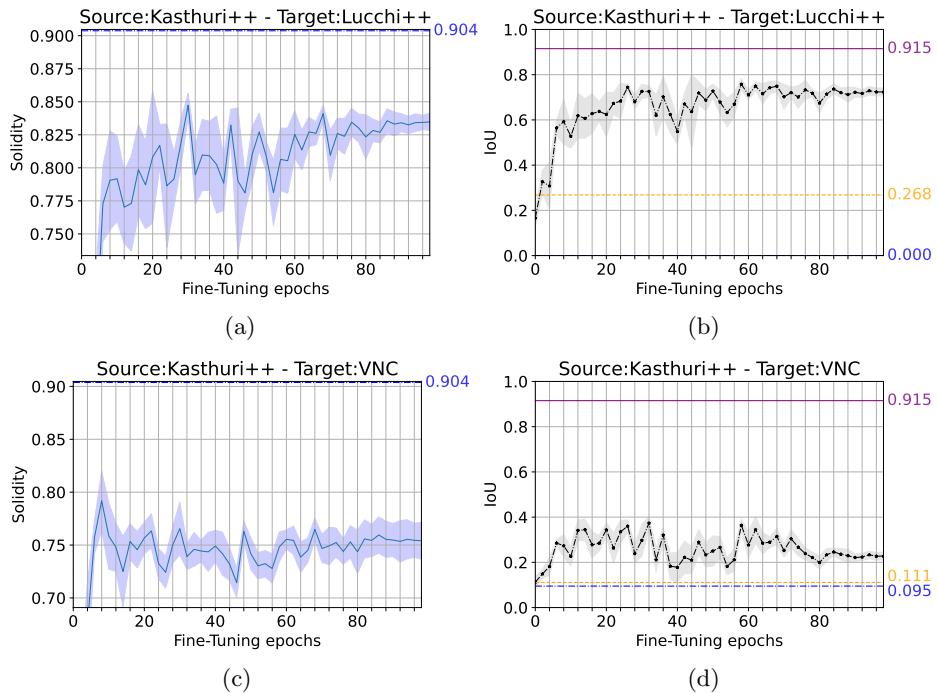


Fig. S2.8: Relation between solidity and IoU in the Attention Y-Net approach with Kasthuri++ as source domain. On the left, the evolution of the solidity value (averaged for ten executions) as a function of the stylization epochs with (a) Lucchi++ and (c) VNC as target domains (dashed lines represent the source solidity value). On the right, the evolution of the test IoU (also averaged over ten executions) as a function of the epochs with (b) Lucchi++ and (d) VNC as target domains. The magenta lines represent the maximum IoU value obtained by the fully supervised baseline models. In contrast, the blue and orange lines represent the IoU values obtained by the baseline methods applied without adaptation and after histogram matching to the target datasets, respectively.

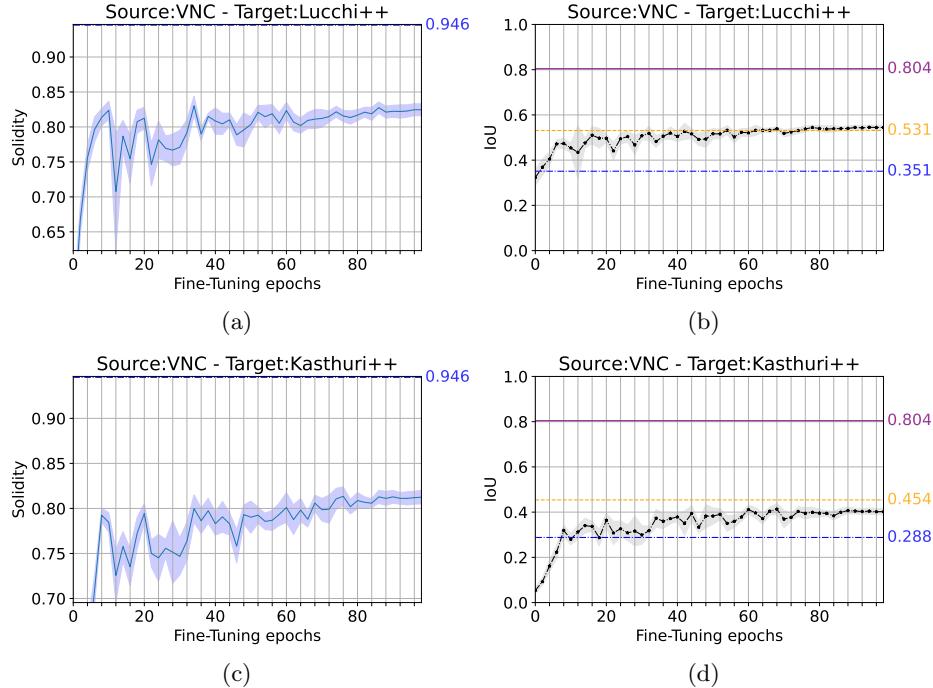


Fig. S2.9: Relation between solidity and IoU in the Attention Y-Net approach with VNC as source domain. On the left, the evolution of the solidity value (averaged for ten executions) as a function of the stylization epochs with (a) Lucchi++ and (c) Kasthuri++ as target domains (dashed lines represent the source solidity value). On the right, the test IoU evolution (averaged over ten executions) as a function of the epochs with (b) Lucchi++ and (d) Kasthuri++ as target domains. The magenta lines represent the maximum IoU value obtained by the fully supervised baseline models, while the blue and orange lines represent the IoU values obtained by the baseline methods applied without adaptation and after histogram matching to the target datasets, respectively.

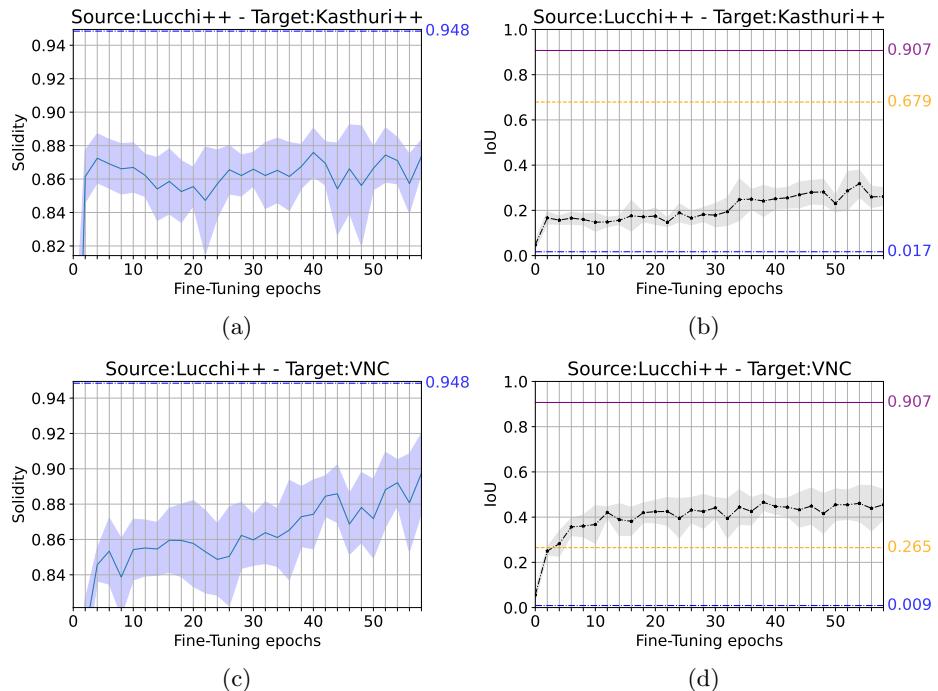


Fig. S2.10: Relation between solidity and IoU in the DAMT-Net approach with Lucchi++ as source domain. On the left, the evolution of the solidity value (averaged for ten executions) as a function of the stylization epochs with (a) Kasthuri++ and (c) VNC as target domains (dashed lines represent the source solidity value). On the right, the evolution of the test IoU (also averaged over ten executions) as a function of the epochs with (b) Kasthuri++ and (d) VNC as target domains. The magenta lines represent the maximum IoU value obtained by the fully supervised baseline models. In contrast, the blue and orange lines represent the IoU values obtained by the baseline methods applied without adaptation and after histogram matching to the target datasets, respectively.

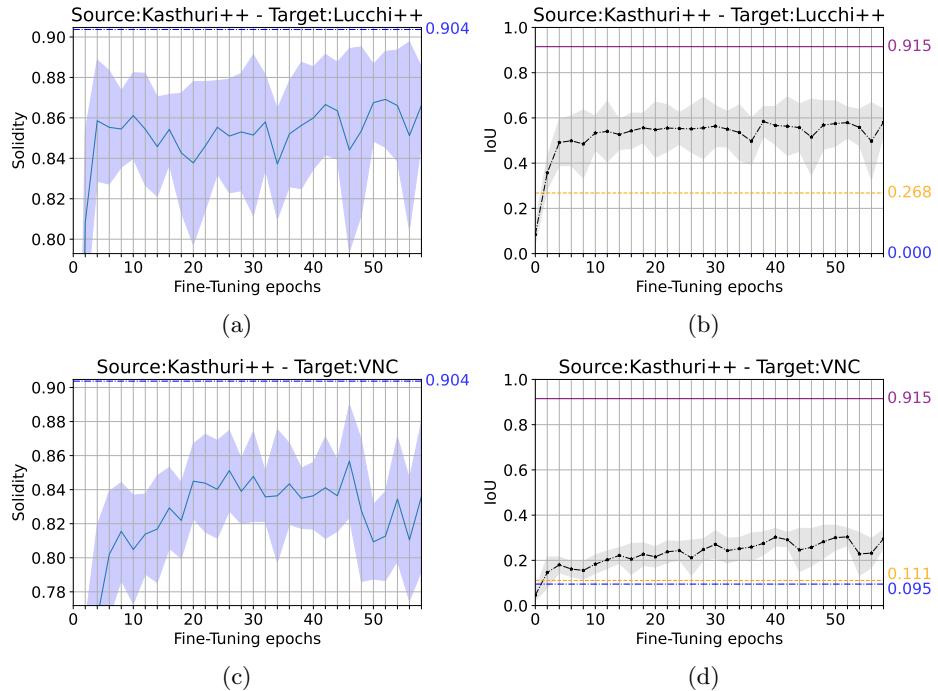


Fig. S2.11: Relation between solidility and IoU in the DAMT-Net approach with Kasthuri++ as source domain. On the left, the evolution of the solidility value (averaged for ten executions) as a function of the stylization epochs with (a) Lucchi++ and (c) VNC as target domains (dashed lines represent the source solidility value). On the right, the evolution of the test IoU (also averaged over ten executions) as a function of the epochs with (b) Lucchi++ and (d) VNC as target domains. The magenta lines represent the maximum IoU value obtained by the fully supervised baseline models. In contrast, the blue and orange lines represent the IoU values obtained by the baseline methods applied without adaptation and after histogram matching to the target datasets, respectively.

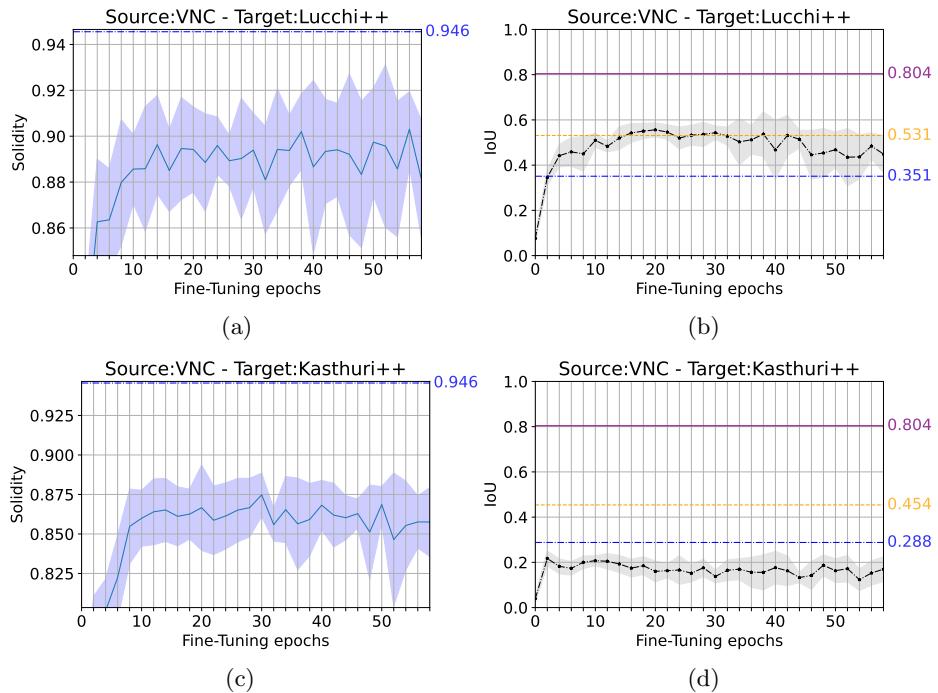


Fig. S2.12: Relation between solidity and IoU in the DAMT-Net approach with VNC as source domain. On the left, the evolution of the solidity value (averaged for ten executions) as a function of the stylization epochs with (a) Lucchi++ and (c) Kasthuri++ as target domains (dashed lines represent the source solidity value). On the right, the test IoU evolution (averaged over ten executions) as a function of the epochs with (b) Lucchi++ and (d) Kasthuri++ as target domains. The magenta lines represent the maximum IoU value obtained by the fully supervised baseline models. In contrast, the blue and orange lines represent the IoU values obtained by the baseline methods applied without adaptation and after histogram matching to the target datasets, respectively.

S3 Hyperparameter search

This section describes in detail the search we performed for the optimal training configuration and set of hyperparameters in all our proposed approaches. The corresponding search space and best values are summarized in the tables below using the following notation:

- $[a, b]$: Range between two possible values. E.g. $\text{zoom}([0.75, 1.25])$ corresponds to a random zoom value between 0.75 and 1.25.
- $[a, b, c]$: All values from a to b with c step. E.g. $[10, 300, 10]$ corresponds to 10, 20, 30, 40, ..., 300.
- (a, b, c) : All values set, e.g., $\text{dropout}(0.1, 0.2, 0.3)$ in a 3-depth level network indicate that 0.1 dropout value has been set in the first level, 0.2 dropout in the second level and 0.3 in the third level.
- $\text{choice}[a, b, \dots]$: One value between a , b and so on. E.g. $[10, 15, 20, 30, 60]$ possible values are: 10 or 15 or 20 or 30 or 60 (but only one).
- a, b, c, \dots : All tested values, e.g., flips, rotations.
- $N(\mu, \sigma)$: normal distribution with mean μ and standard deviation σ .

There are also acronyms used in tables:

- MAE: Mean absolute error.
- MSE: Mean squared error.
- BCE: Binary cross entropy.
- SGD: Stochastic gradient descent.
- Reduce on plateau: A learning rate policy to reduce the value of the learning rate when the monitored metric has stopped improving (https://keras.io/api/callbacks/reduce_lr_on_plateau/).
- OneCycle: One-cycle learning rate policy for super-convergence [1].
- ELU: Exponential linear unit activation function.
- he_normal: *He normal* [2] as kernel initialization.

S3.1 Self-supervised approach

Hyperparameter	Search space	Best assignment
Data		
Validation	True	True
Random validation	True	True
% of train as validation	10%	10%
Patches	Maximum number that fitted into the slice dimensions	Maximum number that fitted into the slice dimensions
Patch size	256×256	256×256
Discard patches with less than a % of the foreground class	<i>choice</i> [True(2%), True(5%), True(10%), False]	True(2%)
Shuffle train on each epoch	True	True
Data augmentation	flips, rotation_range([-180,180,90])	flips, rotation_range([-180,180,90])
Number of epochs	<i>choice</i> [100,200,300]	200
Batch size	<i>choice</i> [1, 2, 5]	1
Loss type	<i>choice</i> [MAE,MSE]	MAE
Optimizer	<i>choice</i> [SGD, Adam]	Adam
Learning rate	<i>choice</i> [1e-3, 2e-3, 1e-4, 5e-4]	5e-4
Scheduler	<i>choice</i> [Reduce on Plateau, OneCycle]	OneCycle
Patience	<i>choice</i> [None,3, 5, 7, 10, 50]	None
Architecture		
Number of feature maps to start with	32	32
Dropout type	Spatial dropout (0.1, 0.1, 0.2, 0.2, 0.3)	Spatial dropout (0.1, 0.1, 0.2, 0.2, 0.3)
Pooling type	Max-pooling	Max-pooling
Kernel initializer	he_normal	he_normal
Activation	ELU	ELU

Table S3.1: Hyperparameter search space for the proposed self-supervised learning method.

Hyperparameter	Search space	Best assignment
Data		
Validation	True	True
Random validation	True	True
% of train as validation	10%	10%
Patches	Maximum number that fitted into the slice dimensions 256×256	Maximum number that fitted into the slice dimensions 256×256
Patch size		
Discard patches with less than a % of the foreground class	<i>choice</i> [True(2%), True(5%), True(10%), False]	True(2%)
Shuffle train on each epoch		
Data augmentation	True	True
Number of epochs	flips, rotation_range([-180,180,90])	flips, rotation_range([-180,180,90])
Batch size	<i>choice</i> [10,200,10] <i>choice</i> [1, 2, 5]	60
Loss type	BCE	1
Optimizer	<i>choice</i> [SGD, Adam]	BCE
Learning rate	<i>choice</i> [1e-3, 2e-3, 1e-4]	Adam
Scheduler	<i>choice</i> [Reduce on Plateau, OneCycle]	1e-4
Patience	<i>choice</i> [None, 3, 5, 7, 10, 50]	OneCycle
Architecture		
Number of feature maps to start with	32	32
Dropout type	Spatial dropout (0.1, 0.1, 0.2, 0.2, 0.3)	Spatial dropout (0.1, 0.1, 0.2, 0.2, 0.3)
Pooling type	Max-pooling	Max-pooling
Kernel initializer	he_normal	he_normal
Activation	ELU	ELU

Table S3.2: Hyperparameter search space for the proposed self-supervised learning training step with the Attention U-Net.

S3.2 Attention Y-Net

Hyperparameter	Search space	Best assignment
Data		
Validation	True	True
Random validation	True	True
% of train as validation	10%	10%
Patches	<i>choice</i> [Sequential, Random(1000)]	Random(1000)
Patch size	256×256	256×256
Discard patches with less than a % of the foreground class	<i>choice</i> [True(5%), True(10%), False]	False
Discard patches with more than a % of zeros in the image	<i>choice</i> [True(50%), True(80%), False]	True(50%)
Shuffle train on each epoch	True	True
Data augmentation	flips, rotation_range(180)	flips, rotation_range(180)
Number of epochs	<i>choice</i> [20, 30, 50, 360]	50
Batch size	<i>choice</i> [1, 2]	1
Loss type	$\alpha MSE + (1 - \alpha) BCE$	$\alpha MSE + (1 - \alpha) BCE$
loss weight α	<i>choice</i> [0.9, 0.98]	0.98
Optimizer	<i>choice</i> [SGD, Adam]	SGD
Learning rate	<i>choice</i> [1e-3, 2e-3]	1e-3
Scheduler	<i>choice</i> [Reduce on Plateau, OneCycle]	Reduce on Plateau
Patience	<i>choice</i> [3, 5, 7, 10, 50]	7
Architecture		
Number of feature maps to start with	<i>choice</i> [32, 64]	32
Dropout type	Spatial dropout	Spatial dropout
Pooling type	(0.1, 0.1, 0.2, 0.2, 0.3)	(0.1, 0.1, 0.2, 0.2, 0.3)
Kernel initializer	Max-pooling	Max-pooling
Activation	he_normal	he_normal
	ELU	ELU

Table S3.3: Hyperparameter search space for the proposed Attention Y-Net, first multitask step.

Hyperparameter	Search space	Best assignment
Data		
Validation	True	True
Random validation	True	True
% of train as validation	10%	10%
Patches	<i>choice</i> [Sequential, Random(1000)]	Random(1000)
Patch size	256 × 256	256 × 256
Discard patches with less than a % of the foreground class	<i>choice</i> [True(5%), True(10%), False]	False
Discard patches with more than a % of zeros in the image	<i>choice</i> [True(50%), True(80%), False]	True(50%)
Shuffle train on each epoch	True	True
Data augmentation	flips, rotation_range(180), histogram_matching(0%, 25%, 50%, 100%), CLAHE(50%)	flips, rotation_range(180), histogram_matching(50%)
Number of epochs	<i>choice</i> [15, 20, 30, 40]	40
Batch size	1	1
Loss type	MSE	MSE
Optimizer	<i>choice</i> [SGD, Adam]	Adam
Learning rate	<i>choice</i> [2e-3, 2e-4]	2e-4
Scheduler	<i>choice</i> [Reduce on Plateau, OneCycle]	Reduce on Plateau
Patience	<i>choice</i> [5, 6, 7, 10, 20]	6
Architecture		
Number of feature maps to start with	32	32
Dropout type	Spatial dropout (0.1, 0.1, 0.2, 0.2, 0.3)	Spatial dropout (0.1, 0.1, 0.2, 0.2, 0.3)
Pooling type	Max-pooling	Max-pooling
Kernel initializer	he_normal	he_normal
Activation	ELU	ELU

Table S3.4: Hyperparameter search space for the proposed Attention Y-Net, second step focused in reconstruction.

Hyperparameter	Search space	Best assignment
Data		
Validation	True	True
Random validation	True	True
% of train as validation	10%	10%
Patches	<i>choice</i> [Sequential, Random(160, 500, 1000, 2000)]	Random(1000)
Patch size	256 × 256	256 × 256
Discard patches with less than a % of the foreground class	<i>choice</i> [True(5%), True(10%), False]	False
Discard patches with more than a % of zeros in the image	<i>choice</i> [True(50%), True(80%), False]	True(50%)
Shuffle train on each epoch	True	True
Data augmentation	flips, rotation_range(180), histogram_matching(0%, 25%, 50%, 100%), CLAHE(50%)	flips, rotation_range(180), CLAHE(50%)
Number of epochs	<i>choice</i> [5, 10, 20, 100]	100
Batch size	1	1
Loss type	BCE	BCE
Optimizer	<i>choice</i> [SGD, Adam]	Adam
Learning rate	<i>choice</i> [2e-3, 2e-4]	2e-4
Scheduler	<i>choice</i> [Reduce on Plateau, OneCycle, None]	oneCycle
Patience	<i>choice</i> [10, 15, 20, 150]	15
Architecture		
Number of feature maps to start with	32	32
Dropout type	Spatial dropout {0.1, 0.1, 0.2, 0.2, 0.3}	Spatial dropout {0.1, 0.1, 0.2, 0.2, 0.3}
Pooling type	Max-pooling	Max-pooling
Kernel initializer	he_normal	he_normal
Activation	ELU	ELU

Table S3.5: Hyperparameter search space for the proposed Attention Y-Net, third and last training step, focused in segmentation.

S3.3 DAMT-Net

To execute DAMT-Net, we follow the publicly available implementation provided by its authors [3]. Since they use two images in each training step, we interpret the batch size as 2. Bearing this in mind, we define an epoch as follows:

$$\text{Epoch} = \frac{|X_{train}|}{batch_size} \quad (1)$$

where $|X_{train}|$ is the cardinality of the training set. Taking this into account, we explored a few hyperparameters:

- Patch size: **512**×**512** pixels and 256 × 256 pixels
- Epochs: 30, **60**, 100
- Save checkpoint every **2** epochs.

Among the different options, the best assignment has been highlighted in bold. The rest of the parameters are those proposed by default in the original publication [3].

References

1. L. N. Smith and N. Topin, “Super-convergence: very fast training of neural networks using large learning rates,” in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, T. Pham, Ed., vol. 11006, International Society for Optics and Photonics. SPIE, 2019, pp. 369 – 386. [29](#)
2. K. He, X. Zhang, S. Ren, and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034. [29](#)
3. J. Peng, J. Yi, and Z. Yuan, “Unsupervised mitochondria segmentation in EM images via domain adaptive multi-task learning,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 6, pp. 1199–1209, 2020. [35](#)