# Supplemental Document for Deformation-aware Unpaired Image Translation for Pose Estimation on Laboratory Animals

Siyuan Li[1], Semih Günel[1,3], Mirela Ostrek[1], Pavan Ramdya[3], Pascal Fua[1], and Helge Rhodin[1,2]

[1]CVLAB, EPFL, Lausanne
[2]Imager Lab, UBC, Vancouver
[3]Neuroengineering Lab, EPFL, Lausanne

In this document, we supply additional evaluation, training, and implementation details, and provide a more details on the ablation study. The stability of the generated images is shown at hand of a short supplemental video.

## 1. Additional qualitative results

We included only few qualitative experiments in the main document due to space constraints. Fig. 2 provides additional examples of the image generation quality and the accuracy of the associated keypoint annotations, inferred via our explicit deformation field.

Moreover, Fig. 3 shows additional examples of the pose estimation quality compared to using Cycle-GAN. Our approach produces much fewer miss classifications, for instance, in the case of extreme bending positions of the worm.

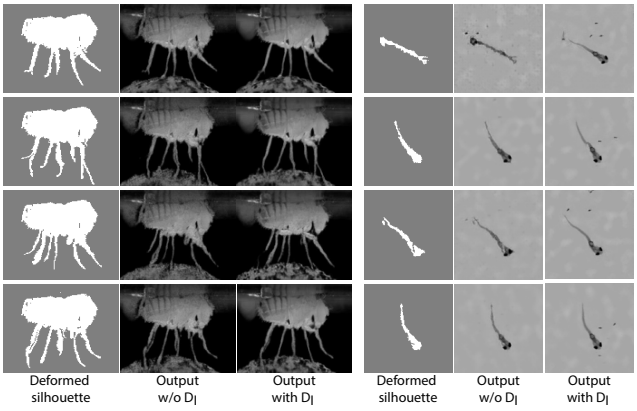


Figure 1. **Ablation study on $D_I$.** Without $D_I$, small artifacts in the generated (deformed) segmentation masks lead to unrealistic images.

## 2. Ablation study details

The ablation study in the main document tests our complete approach while removing of our core contributions in terms of the PCK metric at threshold 15 pixels. The additional metrics in Table 1 show that our contributions improve consistently across different PCK thresholds. Each of our contributions is significant with gains of 7 to 25 on PCK-5 and 1 to 12 AUC points. Notably, using global affine deformation is worse than without any deformation. This may be because the affine network rotates the body of synthetic fly to match the shape of real fly. However, the rotation also affects the leg orientation, which leads to less realistic poses. It is best to use global and local motion together (Ours).

We also experimented with using classical domain adaptation techiques, see details explained in Section 5. However, we could not make these methods developed for classification tasks work on the regression task of 2D pose estimation. Table 1 shows that the best ADDA variant does not even outperform training on synthetic images. None of the variants can translate between the large appearance and pose and shape gaps between source and target. While future work may improve on this, it shows that straight applications of existing domain adaptation techniques does not suffice.

Moreover, Fig. 1 provides additional results comparing the generated image quality with and without using $D_I$. Clear improvements are gained for the fish and Drosophila. For instance, legs are properly superimposed on the ball, while holes arise without $D_I$ (therefore, without end-to-end training). No significant improvement could be observed on the worm case due to its simplicity.

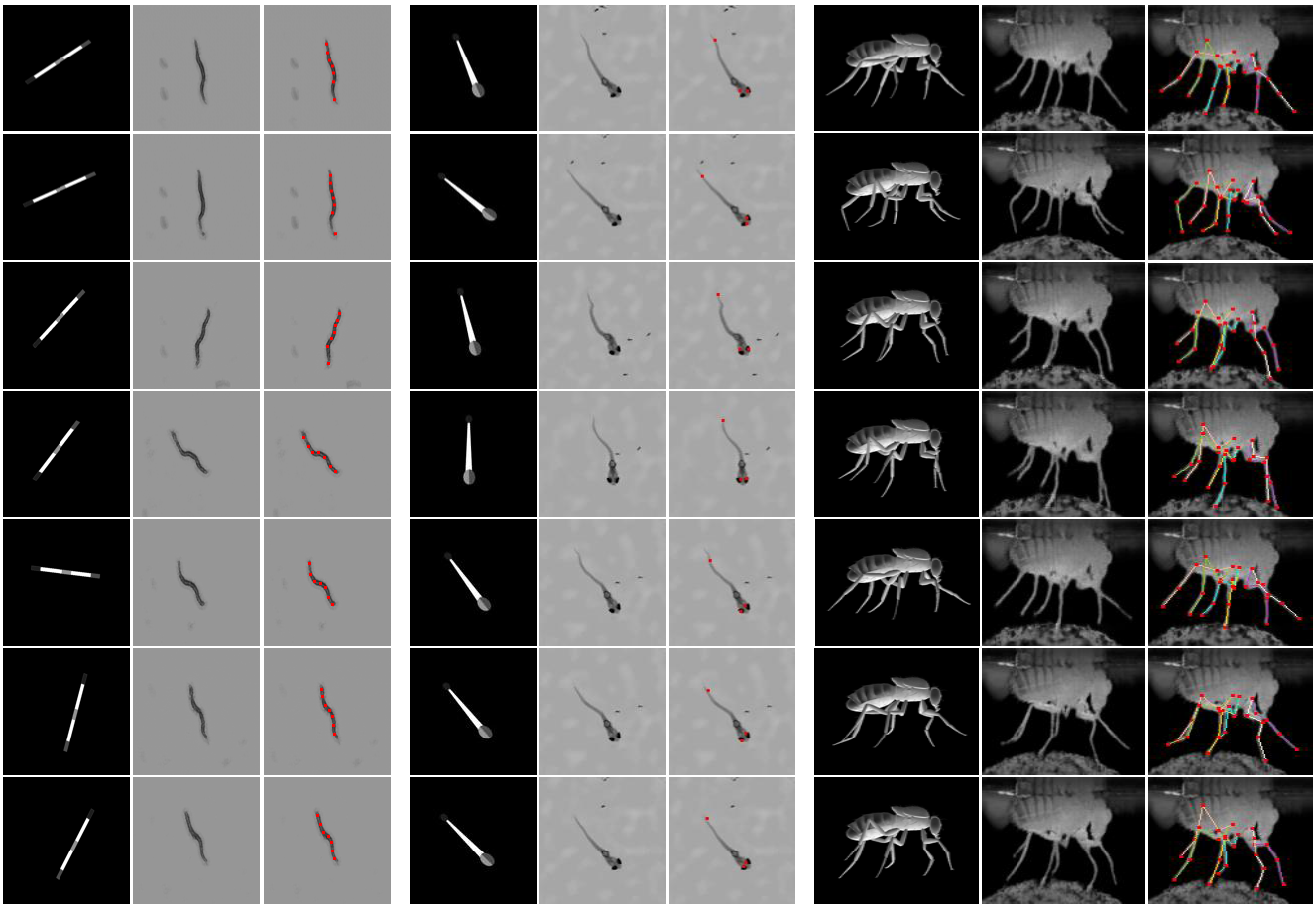


Figure 2. **Qualitative image generation results.** Our approach can generate realistic and diverse poses, which are transferred across domains faithfully. Our method works on all three tested animals, including the *Drosophila* dataset with superimposed legs that are on the ball that has no correspondence in the source domain.

| Metric | learning rate decay | PI-PCK ↑ (5 pix) | PI-PCK ↑ (15 pix) | PI-AUC ↑ (4-45 pix) |
|---|---|---|---|---|
| **Ours** | disabled | **40.0** | **84.7** | **86.0** |
| Ours with lr decay | enabled | 38.6 | 83.2 | 85.1 |
| Ours w/o global deformation | enabled | 31.4 | 79.2 | 84.0 |
| Ours w/o deformation | enabled | 18.5 | 64.9 | 74.9 |
| Ours w/o local deformation | enabled | 13.1 | 57.4 | 73.8 |
| Ours using vector field | enabled | 18.6 | 69.1 | 79.0 |
| Synthetic | n/a | 19.8 | 67.9 | 75.75 |
| ADDA | n/a | 7.63 | 55.5 | 66.3 |

Table 1. **Detailed ablation study on *Drosophila Melanogaster*.** All model components contribute to the final reconstruction accuracy. The learning rate (lr) decay means using learning rate decay during training the deformation and appearance transfer network.

## 3. Dataset sources and splits

The worm dataset stems from the OpenWorm initiative [8, 7]. We used three videos after subsampling to 8x speed. The OpenWorm videos are referred by strain type and timestamp. We used the three videos specified in Table 2, downloaded from YouTube at subsampled framerate (8x speed compared to the original recording).

| Strain | Strain description | Time stamp |
|---|---|---|
| OW940 | zgIs128[P(dat-1)::alpha-Synuclein::YFP] | 2014-03-14T13:39:36+01:00 |
| OW940 | zgIs128[P(dat-1)::alpha-Synuclein::YFP] | 2014-03-06T09:11:51+01:00 |
| OW939 | zgIs113[P(dat-1)::alpha-Synuclein::YFP] | 2014-02-22T14:13:49+01:00 |

Table 2. **OpenWorm videos.** Strain type and timestamp of the used videos published by [8, 7].

The worm is tracked in each video to be roughly centered. The only transformation done is scaling the original frames to resolution $128 \times 128$ pixels. We randomly picked 100 frames of these three videos for test and then picked 1000 frames out of all remaining frames for unpaired training. We manually annotated every 10th frame (100 frames) from the unpaired training examples with two

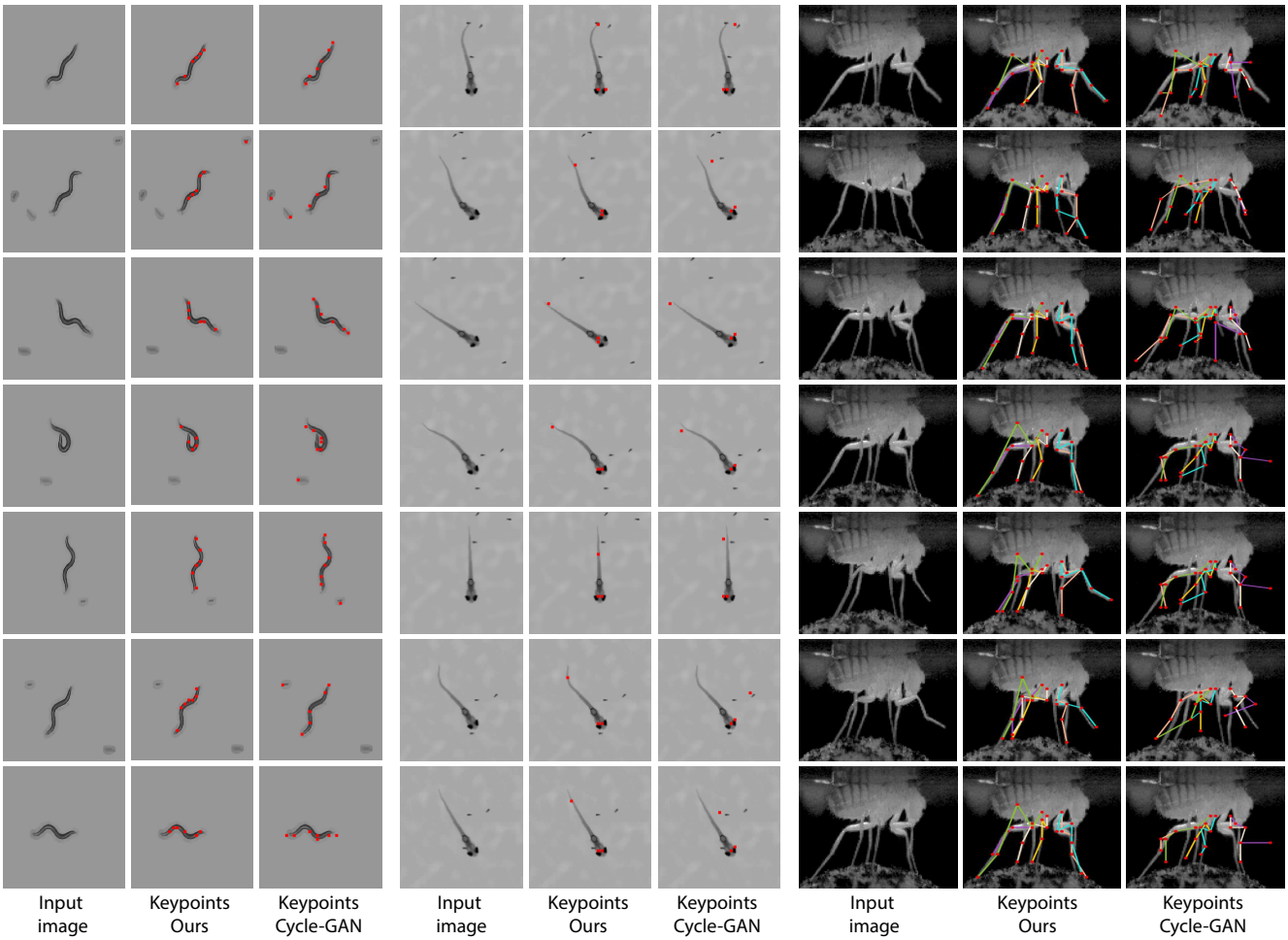


Figure 3. **Pose estimation result comparison.** Training a pose estimator on our generated images yields accurate detections with far less failures when compared to Cycle-GAN, the best performing baseline. The *Drosophila* case is most challenging as the legs are thin and self-similar.

keypoints (head and tail) to train the supervised baseline, and the entire test set (100 frames) for quantifying pose estimation accuracy.

For the zebrafish larva experiments, we used Video 3 (672246_file04.avi) published in the supplemental of [**?**] (biorxiv.org/content/10.1101/672246v1.supplementary-material). We crop the original video from $1920 \times 1080$ pixels to the region with top left corner $(500, 10)$ and bottom right $(1290, 800)$, and scale it to $128 \times 128$ pixels. We deleted some repetitive frames where the zebrafish is not moving to increase the percentage of frames where zebrafish is bending. In total, we retained 600 frames. We selected the last 100 frames for test and 500 left for unpaired training. Besides the test images, we also manually annotated every 5th (100 frames) from the 500 training images as the training data for the supervised baseline.

## 4. Training details

**Training The Unpaired Image Translation Network.** We use the Adam optimizer with different initial learning rates for different modules. For $G_I$, $D_I$, we set the learning rate to $2\mathrm{e}{-3}$. For $G_S$, we set the learning rate to $2\mathrm{e}{-5}$ since a slight update will have a big impact on the deformation field due to the integrating the spatial gradient in the last layer of $G_S$. We set the learning rate of $D_S$ to $1/10$ the one of $G_S$, which balanced the influence of $G_S$ and $D_S$ in our experiments. In case of *Drosophila* training, we trained the networks for 30 epoch with fixed learning rate. We select the weights for $G_S$ and $G_I$ at 14th epoch. We also do experiments with linearly decayed learning rate. We start the decay the learning rate at epoch 50 and reduce it to 0 till epoch 100. It shows that very small learning rate will introduce artifacts on the deformed mask and decrease the performance a little. For fish and worm, we set the learning rate of $G_I$ and $D_I$ to $2\mathrm{e}-4$, $G_S$ to $2\mathrm{e}-5$, to account for the

simpler setting of deforming from a single template image. Additionally, we set learning rate for $D_S$ to be $1/100$ to the one of $G_S$ for fish to better balance the training of $G_S$ and $D_S$. We trained the networks for 100 epochs and selected the weights for $G_S$ and $G_I$ at 70th epochs for worm and 100th for fish.

We set $\alpha$ to be $2\mathrm{e}-3$ and $\beta$ to be $2\mathrm{e}-2$ in the regularization terms, and we set $\lambda$ to be $1e-5$ for fly and worm and $1e-4$ for fish in $\mathcal{L}_I$.

The batch size of the image translation training is set to 4. An other important detail is the initialization of $G_S$ to generate the identity mapping. We achieved that by initially training $G_S$ solely on the regularization term, which pushes it towards this state.

**Training Pose Estimation Network.** We use Adam optimizer with initial learning rate of $2\mathrm{e}{-4}$. We train the pose estimation network for 200 epochs and the learning rate starts to linear decay after epoch 100, till epoch 200. Input images are augmented by random rotations, drawn uniformly from $[-30°, 30°]$ for *Drosophila*.

## 5. Implementation details

**Deformation representation.** Directly modeling the deformation as vector field will make the transformation unstable and easily lose the semantic correspondence. For example, a vector field permits coordinate crossing and disconnected areas, which leads to unstable training and divergence. In order to preserve a connected grid topology, we model our deformation close to the difformorphic transformation, which generates the deformation field as the integral of a velocity field. This leads to useful properties such as invertibility and none crossing intersections [1]. However, it is in general expensive to compute the integral over an axis, thus making it difficult to incorporate into deep networks. Instead of modeling a continuous velocity function, we directly model our deformation field $\phi$ as the integral of the spatial gradient of vector field, as proposed by Shu et al. [9]. We write,

$$\bigtriangledown \phi_x = \frac{\partial \phi}{\partial x} \qquad \bigtriangledown \phi_y = \frac{\partial \phi}{\partial y} \tag{1}$$

where $x$, $y$ define the gradient directions along the image axes. The $\phi_x$ and $\phi_y$ measure the difference of consecutive pixels. By enforcing the difference to be positive (e.g., by using ReLU activation functions; we use HardTanh with range (0, 0.1)), we avoid self-crossing and unwanted disconnected areas. For example, when $\phi_x$ and $\phi_y$ equals to 1, the distance between the consecutive pixels is the the same. If $\phi_x, \phi_y > 1$ , the distance will increase, otherwise, when $\phi_x$ ,$\phi_y < 1$, it will decrease.

The second module is the spatial integral layer, also the last layer of deformation spatial gradient generator. This layer sums the spatial gradients along the x and y directions and produces the final deformation field,

$$\phi_{i,j} = (\sum_{m=0}^{i} \bigtriangledown \phi_{x_m}, \sum_{n=0}^{j} \bigtriangledown \phi_{y_n}), \tag{2}$$

where $i, j$ is the pixel location. Since the $u$, $v$ in general position do not correspond to one exact pixel location in the source image, we compute the output image using a differentiable bilinear interpolation operation, as for spatial transformers [4].

**Shape Discriminator** We utilize the $70 \times 70$ patchGAN discriminator as our backbone structure [3] . The patch-wise design makes the network focus on the local area of the shape. Furthermore, if the shape between two domains is extremely different, the patch-wise design prevents the discriminator from converging too quickly. However, the design also limits the network's awareness of global shape changes [2]. Thus, we add dilation to the second and the third convolution layers of patchGAN. Those dilated layers enlarge the receptive field of our shape discriminator, making it aware of bigger shape variation, giving a better guidance to the generator.

**Image Generator.** We build our generator on the U-Net architecture, which is proved to be effective in tasks such as pixel-wise image translation and segmentation [6]. The generator contains several fully convolutional down-sampling and up-sampling layers. The skip connections in the generator help to propagate information directly from input features to the output, which guarantee the preservation of spatial information in the output image.

**Pose Estimator.** We adopt the stacked hourglass human pose estimation network to perform pose estimation on animals [5]. The stacked hourglass network contains several repeated bottom-up, top-down processing modules with intermediate supervision between them. A single stack hourglass module consists of several residual bottleneck layers with max-pooling, following by the up-sampling layers and skip connections. We used 2 hourglass modules in our experiments. The pose estimation network is trained purely on the animal data we generated; without pre-training and manually annotated labels. The ground-truth poses come from the annotations of synthetic animal models. The pose invariant (PI) training is performed in all experiments labeled with *PI training*.

**Pose Annotation.** *Drosophila* has six limbs, each limb has five joints, giving 30 2D keypoints that we aim to detect. By using our image translation model, we generated 1500 images with annotation from the synthetic data. Each image is in size $128 \times 128$ pixels. The first hourglass network is preceded with convolutional layers that reduce the input

image size from $128 \times 128$ to $32 \times 32$. The second hourglass does not change the dimension. Thus, the network will output a $30 \times 32 \times 32$ tensor, which represents the probability maps of 30 different joints locations. For training, we create the ground truth label using a 2D Gaussian with mean at the annotated keypoint and 0.5 on the diagonal of the covariance matrix. The training loss is the MSE between the generated probability map and the ground truth label.

We annotated three keypoints on D. rerio and seven keypoints on C. elegans. We use the same network as for Drosophila, but the output tensor adapted to the number of keypoints, $3 \times 32 \times 32$ and $7 \times 32 \times 32$, respectively.

**Domain Adaptation: ADDA** We adapted the ADDA pipeline to make it work for pose estimation problem. We use two same hourglass networks with 2 sub-hourglass stacks [5] as source and target domain feature extractor. The feature extraction network takes an image and output a spatial feature map $(N \times C \times H \times W)$. C is 256 and H,W are 32 and 32 respectively if the size of input images is $128 \times 128$. We use 3 additional convolutional layers with BatchNorms and ReLU to perform pose estimation from the learned feature maps. For the discriminator, we use the normal 3 layer PatchGAN [3]. We follow the training pipeline of ADDA. At first, we train the source feature extractor and pose estimator for 200 epochs. Then, we fix their weights and start the adversarial training for target feature extractor and the discriminator which tries to distinguish between source and target features. We set the learning rate to 2e-4 and start to decay from 100 epoch till 200 epoch for both tasks. After training, we select the weight of the 190th epoch, based on the validation results.

## References


[1] John Ashburner. A fast diffeomorphic image registration algorithm. *Neuroimage*, 38(1):95–113, 2007. 4

[2] Aaron Gokaslan, Vivek Ramanujan, Daniel Ritchie, Kwang In Kim, and James Tompkin. Improving shape deformation in unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 649–665, 2018. 4

[3] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017. 4, 5

[4] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial Transformer Networks. In *NeurIPS*, pages 2017–2025, 2015. 4

[5] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked Hourglass Networks for Human Pose Estimation. *ECCV*, pages 483–499, 2016. 4, 5

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015. 4

[7] Balazs Szigeti, Padraig Gleeson, Michael Vella, Sergey Khayrulin, Andrey Palyanov, Jim Hokanson, Michael Currie, Matteo Cantarelli, Giovanni Idili, and Stephen D. Larson. Openworm: an open-science approach to modeling caenorhabditis elegans. *Front. Comput. Neurosci.*, 2014. 2

[8] Eviatar Yemini, Tadas Jucikas, Laura J Grundy, André E X Brown, and William R Schafer. A database of caenorhabditis elegans behavioral phenotypes. In *Nature Methods*, 2013. 2

[9] Riza Alp Guler Dimitris Samaras Nikos Paragios Zhixin Shu, Mihir Sahasrabudhe and Iasonas Kokkinos. Deforming autoencoders: Unsupervised disentangling of shape and appearance. In *ECCV*, 2018. 4