**Installation:**

We recommend installing on a Linux-based workstation using a recent model NVIDIA GPU. Install a conda environment for your distro, for example:

https://conda.io/projects/conda/en/latest/user-guide/install/linux.html

Create a new conda environment with python=3.10

conda create -n "embgan" python=3.10.0

conda activate embgan

Install pytorch v1.13.1 and an appropriate version of CUDA for your GPU

conda install pytorch==1.13.1 torchvision==0.14.1 torchaudio==0.13.1 pytorch-cuda=11.6 -c pytorch -c nvidia

Download the embGAN codebase and pretrained model weights

**Training a model from scratch:**

Calling the train.py function inside a training directory uses a simple syntax:

python train.py --cuda --outpath /path/to/output/directory --seed 100

To further randomize starting weights between training runs, the random seed can be changed.

The training script expects a simple directory structure and should be called in the working directory with subfolders:

/data

> train

>dic

>labels

> val

>dic

>labels

Where each the corresponding DIC and label images are saved as 8-bit 2D TIF images with matching filenames.

**Running inference with a trained model:**

Calling the inference.py function uses a similarly simply syntax:

python3 inference.py --out\_path /output/data/path --data\_path /input/data/path --model\_path /path/to/model/weights/final\_weights.pth

Input data should be contrast adjusted and provided as 8-bit 2D TIF images. This can be easily generated from a multidimensional image in Fiji/ImageJ by using the save as → image sequence function. Output images can then be easily reassembled into multidimensional TIF stacks by using the import image sequence function.