



Self-supervised Dense Representation Learning for Live-Cell Microscopy with Time Arrow Prediction

Benjamin Gallusser, Max Stieber, and Martin Weigert^(✉)

École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
{benjamin.gallusser,max.stieber,martin.weigert}@epfl.ch

Abstract. State-of-the-art object detection and segmentation methods for microscopy images rely on supervised machine learning, which requires laborious manual annotation of training data. Here we present a self-supervised method based on *time arrow prediction pre-training* that learns dense image representations from raw, unlabeled live-cell microscopy videos. Our method builds upon the task of predicting the correct order of time-flipped image regions via a single-image feature extractor followed by a time arrow prediction head that operates on the fused features. We show that the resulting dense representations capture inherently time-asymmetric biological processes such as cell divisions on a pixel-level. We furthermore demonstrate the utility of these representations on several live-cell microscopy datasets for detection and segmentation of dividing cells, as well as for cell state classification. Our method outperforms supervised methods, particularly when only limited ground truth annotations are available as is commonly the case in practice. We provide code at <https://github.com/weigertlab/tarrow>.

Keywords: Self-supervised learning · Live-cell microscopy

1 Introduction

Live-cell microscopy is a fundamental tool to study the spatio-temporal dynamics of biological systems [4, 24, 26]. The resulting datasets can consist of terabytes of raw videos that require automatic methods for downstream tasks such as classification, segmentation, and tracking of objects (*e.g.* cells or nuclei). Current state-of-the-art methods rely on supervised learning using deep neural networks that are trained on large amounts of ground truth annotations [6, 25, 31]. The manual creation of these annotations, however, is laborious and often constitutes a practical bottleneck in the analysis of microscopy experiments [6]. Recently, self-supervised representation learning (SSL) has emerged as a promising approach to alleviate this problem [1, 3]. In SSL one first defines a *pretext task*

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-43993-3_52.

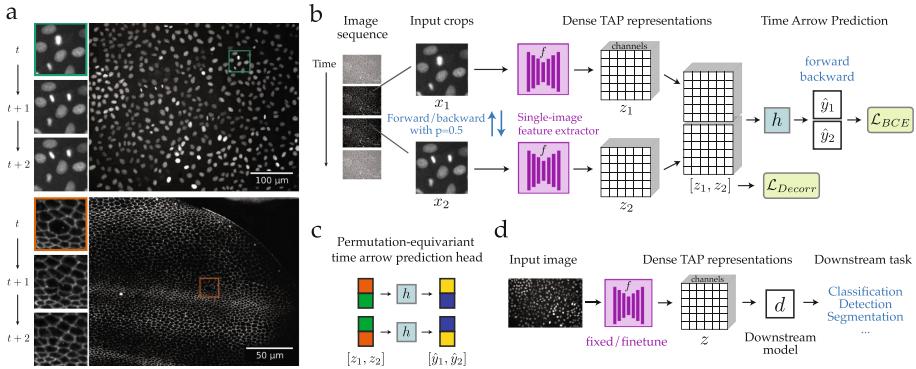


Fig. 1. **a)** Example frames from two live-cell microscopy videos. Top: *MDCK* cells with labeled nuclei [28], Bottom: *Drosophila* wing with labeled membrane [4]. Insets show three consecutive time points containing cell divisions. **b)** Overview of TAP: We create crops (x_1, x_2) from consecutive time points of a given video. After randomly flipping the input order (forward/backward), each crop is passed through a dense feature extractor f creating pixel-wise TAP representations (z_1, z_2). These are stacked and fed to the time arrow prediction head h . **c)** We design h to be permutation-equivariant ensuring consistent classification of temporally flipped inputs. **d)** The learned TAP representations z are used as input to a downstream model d .

which can be formulated solely based on *unlabeled* images (*e.g.* inpainting [8], or rotation prediction [5]) and tasks a neural network to solve it, with the aim of generating latent representations that capture high-level image semantics. In a second step, these representations can then be either *finetuned* or used directly (*e.g.* via *linear probing*) for a *downstream task* (*e.g.* image classification) with available ground truth [7, 10, 18]. Importantly, a proper choice of the pretext task is crucial for the resulting representations to be beneficial for a specific downstream task.

In this paper we investigate whether *time arrow prediction*, *i.e.* the prediction of the correct order of temporally shuffled image frames extracted from live-cell microscopy videos, can serve as a suitable pretext task to generate meaningful representations of microscopy images. We are motivated by the observation that for most biological systems the temporal dynamics of local image features are closely related to their semantic content: whereas static background regions are time-symmetric, processes such as cell divisions or cell death are inherently time-asymmetric (*cf.* Fig. 1a). Importantly, we are interested in *dense* representations of individual images as they are useful for both image-level (*e.g.* classification) or pixel-level (*e.g.* segmentation) downstream tasks. To that end, we propose a time arrow prediction pre-training scheme, which we call TAP, that uses a feature extractor operating on single images followed by a time arrow prediction head operating on the fused representations of consecutive time points. The use of time arrow prediction as a pretext task for natural (*e.g.* youtube) videos was introduced by Pickup *et al.* [19] and has since then seen numerous

applications for image-level tasks, such as action recognition, video retrieval, and motion classification [2, 11, 14, 15, 22, 30]. However, to the best of our knowledge, SSL via time arrow prediction has not yet been studied in the context of live-cell microscopy. Concretely our contributions are: *i*) We introduce the time arrow prediction pretext task to the domain of live-cell microscopy and propose the TAP pre-training scheme, which learns dense representations (in contrast to only image-level representations) from raw, unlabeled live-cell microscopy videos, *ii*) we propose a custom (permutation-equivariant) time arrow prediction head that enables robust training, *iii*) we show via attribution maps that the representations learned by TAP capture biologically relevant processes such as cell divisions, and finally *iv*) we demonstrate that TAP representations are beneficial for common image-level and pixel-level downstream tasks in live-cell microscopy, especially in the low training data regime.

2 Method

Our proposed TAP pre-training takes as input a set $\{I\}$ of live-cell microscopy image sequences $I \in \mathbb{R}^{T \times H \times W}$ with the goal to produce a feature extractor f that generates c -dimensional dense representations $z = f(x) \in \mathbb{R}^{c \times H \times W}$ from single images $x \in \mathbb{R}^{H \times W}$ (*cf.* Fig. 1b for an overview of TAP). To that end, we randomly sample from each sequence I pairs of smaller patches $x_1, x_2 \in \mathbb{R}^{h \times w}$ from the same spatial location but consecutive time points $x_1 \subset I_t, x_2 \subset I_{t+1}$. We next flip the order of each pair with equal probability $p = 0.5$, assign it the corresponding label y (*forward* or *backward*) and compute dense representations $z_1 = f(x_1)$ and $z_2 = f(x_2)$ with $z_1, z_2 \in \mathbb{R}^{c \times h \times w}$ via a fully convolutional feature extractor f . The stacked representations $z = [z_1, z_2] \in \mathbb{R}^{2 \times c \times h \times w}$ are fed to a *time arrow prediction head* h , which produces the classification logits $\hat{y} = [\hat{y}_1, \hat{y}_2] = h([z_1, z_2]) = h([f(x_1), f(x_2)]) \in \mathbb{R}^2$. Both f and h are trained jointly to minimize the loss

$$\mathcal{L} = \mathcal{L}_{BCE}(y, \hat{y}) + \lambda \mathcal{L}_{Decorr}(z), \quad (1)$$

where \mathcal{L}_{BCE} denotes the standard softmax + binary cross-entropy loss between the ground truth label y and the logits $\hat{y} = h(z)$, and \mathcal{L}_{Decorr} is a loss term that promotes z to be decorrelated across feature channels [12, 33] via maximizing the diagonal of the softmax-normalized correlation matrix A_{ij} :

$$\mathcal{L}_{Decorr}(\tilde{z}) = -\frac{1}{c} \log \sum_{i=1}^c A_{ii}, \quad A_{ij} = \text{softmax}(\tilde{z}_i^T \cdot \tilde{z}_j / \tau) = \frac{e^{\tilde{z}_i^T \cdot \tilde{z}_j / \tau}}{\sum_{j=1}^c e^{\tilde{z}_i^T \cdot \tilde{z}_j / \tau}} \quad (2)$$

Here $\tilde{z} \in \mathbb{R}^{c \times 2hw}$ denotes the stacked features z flattened across the non-channel dimensions, and τ is a temperature parameter. Throughout the experiments we use $\lambda = 0.01$ and $\tau = 0.2$. Note that instead of creating image pairs from consecutive video frames we can as well choose a custom time step $\Delta t \in \mathbb{N}$ and sample $x_1 \subset I_t$ and $x_2 \subset I_{t+\Delta t}$, which we empirically found to work better for datasets with high frame rate.

Permutation-equivariant Time Arrow Prediction Head: The time arrow prediction task has an inherent symmetry: flipping the input $[z_1, z_2] \rightarrow [z_2, z_1]$ should flip the logits $[\hat{y}_1, \hat{y}_2] \rightarrow [\hat{y}_2, \hat{y}_1]$. In other words, h should be *equivariant* wrt. to permutations of the input. In contrast to common models (*e.g.* ResNet [9]) that lack this symmetry, we here directly incorporate this inductive bias via a *permutation-equivariant head* h that is a generalization of the set permutation-equivariant layer proposed in [32] to dense inputs. Specifically, we choose $h = h_1 \circ \dots \circ h_L$ as a chain of permutation-equivariant layers h_l :

$$h_l : \mathbb{R}^{2 \times c \times h \times w} \rightarrow \mathbb{R}^{2 \times \tilde{c} \times h \times w}$$

$$h_l(z)_{tmij} = \sigma \left(\sum_n L_{mn} z_{t,n,i,j} + \sum_{s,n} G_{mn} z_{s,n,i,j} \right), \quad (3)$$

with weight matrices $L, G \in \mathbb{R}^{\tilde{c} \times c}$ and a non-linear activation function σ . Note that L operates independently on each temporal axis and thus is trivially permutation equivariant, while G operates on the temporal sum and thus is permutation invariant. The last layer h_L includes an additional global average pooling along the spatial dimensions to yield the final logits $\hat{y} \in \mathbb{R}^2$.

Augmentations: To avoid overfitting on artificial image cues that could be discriminative of the temporal order (such as a globally consistent cell drift, or decay of image intensity due to photo-bleaching) we apply the following augmentations (with probability 0.5) to each image patch pair x_1, x_2 : flips, arbitrary rotations and elastic transformations (jointly for x_1 and x_2), translations for x_1 and x_2 (independently), spatial scaling, additive Gaussian noise, and intensity shifting and scaling (jointly+independently).

3 Experiments

3.1 Datasets

To demonstrate the utility of TAP for a diverse set of specimen and microscopy modalities we use the following four different datasets:

HELa. Human cervical cancer cells expressing histone 2B-GFP imaged by fluorescence microscopy every 30 min [29]. The dataset consists of four videos with overall 368 frames of size 1100×700 . We use $\Delta t = 1$ for TAP training.

MDCK. Madin-Darby canine kidney epithelial cells expressing histone 2B-GFP (*cf.* Fig. 3b), imaged by fluorescence microscopy every 4 min [27, 28]. The dataset consists of a single video with 1200 frames of size 1600×1200 . We use $\Delta t \in \{4, 8\}$.

FLYWING. *Drosophila melanogaster* pupal wing expressing Ecad::GFP (*cf.* Fig. 3a), imaged by spinning disk confocal microscopy every 5 min [4, 20]. The dataset consists of three videos with overall 410 frames of size 3900×1900 . We use $\Delta t = 1$.

YEAST. *S. cerevisiae* cells (*cf.* Fig. 3c) imaged by phase-contrast microscopy every 3 min [16, 17]. The dataset consists of five videos with overall 600 frames of size 1024×1024 . We use $\Delta t \in \{1, 2, 3\}$.

For each dataset we heuristically choose Δt to roughly correspond to the time scale of observable biological processes (*i.e.* larger Δt for higher frame rates).

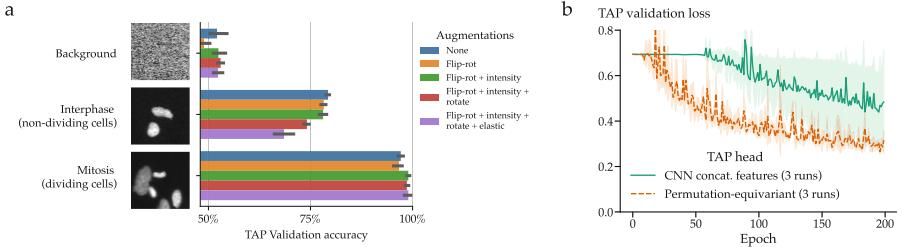


Fig. 2. a) TAP validation accuracy for different image augmentations on crops of background, interphase (non-dividing), and mitotic (dividing) cells (from HE LA dataset). **b)** TAP validation loss during training on FLYWING for a regular CNN time arrow prediction head (green) and the proposed permutation-equivariant head (orange). We show results of three runs per model.

3.2 Implementation Details:

For the feature extractor f we use a 2D U-NET [21] with depth 3 and $c = 32$ output features, batch normalization and leaky ReLU activation (approx. 2M params). The time arrow prediction head h consists of two permutation-equivariant layers with batch normalization and leaky ReLU activation, followed by global average pooling and a final permutation-equivariant layer (approx. 5k params). We train all TAP models for 200 epochs and 10^5 samples per epoch, using the Adam optimizer [13] with a learning rate of 4×10^{-4} with cyclic schedule, and batch size 256. Total training time for a single TAP model is roughly 8h on a single GPU. TAP is implemented in PyTorch.

3.3 Time Arrow Prediction Pretraining

We first study how well the time arrow prediction pretext task can be solved depending on different image structures and used data augmentations. To that end, we train TAP networks with an increasing number of augmentations on HE LA and compute the TAP classification accuracy for consecutive image patches x_1, x_2 that contain either background, interphase (non-dividing) cells, or mitotic (dividing) cells. As shown in Fig. 2a, the accuracy on background regions is approx. 50% irrespective of the used augmentations, suggesting the absence of predictive cues in the background for this dataset. In contrast, on regions with cell divisions the accuracy reaches almost 100%, confirming that

TAP is able to pick up on strong time-asymmetric image features. Interestingly, the accuracy for regions with non-dividing cells ranges from 68% to 80%, indicating the presence of weak visual cues such as global drift or cell growth. When using more data augmentations the accuracy decreases by roughly 12% points, suggesting that data augmentation is key to avoid overfitting on confounding cues.

Next we investigate which regions in full-sized videos are most discriminative for TAP. To that end, we apply a trained TAP network on consecutive full-sized frames x_1, x_2 and compute the dense attribution map of the classification logits y wrt. to the TAP representations z via Grad-CAM [23]. In Fig. 3 we show example attribution maps on top of single raw frames for three different datasets. Strikingly, the attribution maps highlight only a few distributed, yet highly localized image regions. When inspecting the top six most discriminative regions and their temporal context for a single image frame, we find that virtually all of them contain cell divisions (*cf.* Fig. 3). Moreover, when examining the attribution maps for full videos, we find that indeed most highlighted regions correspond to mitotic cells, underlining the strong potential of TAP to reveal time-asymmetric biological phenomena from raw microscopy videos alone (*cf.* Supplementary Video 1).

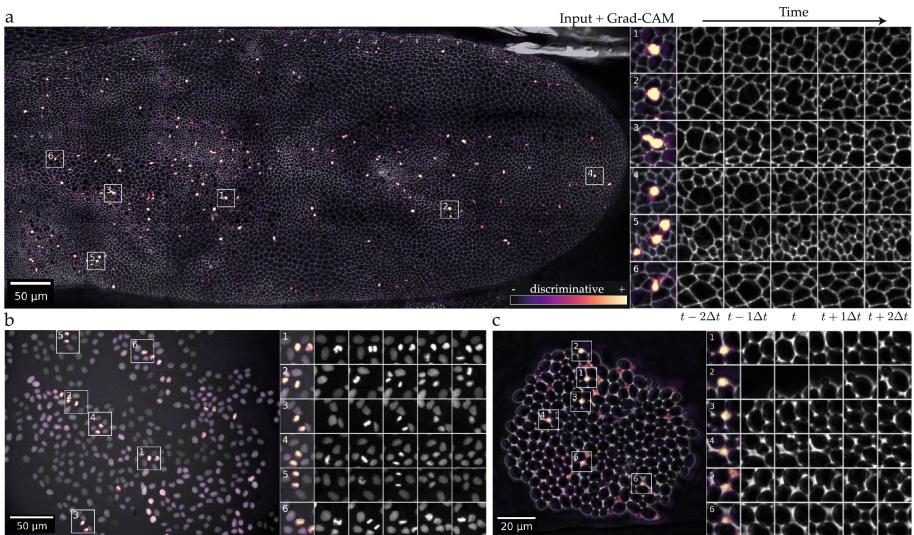


Fig. 3. A single image frame overlayed with TAP attribution maps (computed with Grad-CAM [23]) for a) FLYWING, b) MDCK, and c) YEAST. Insets show the top six most discriminative regions and their temporal context (± 2 timepoints). Note that across all datasets almost all regions contain cell divisions. Best viewed on screen.

Finally, we emphasize the positive effect of the permutation-equivariant time arrow prediction head on the training process. When we originally used a regular

CNN-based head, we consistently observed that the TAP loss stagnated during the initial training epochs and decreased only slowly thereafter (*cf.* Fig. 2b). Using the permutation-equivariant head alleviated this problem and enabled a consistent loss decrease already from the beginning of training.

3.4 Downstream Tasks

We next investigate whether the learned TAP representations are useful for common supervised downstream tasks, where we especially focus on their utility in the low training data regime. First we test the learned representations on two image-level classification tasks, and later on two dense segmentation tasks.

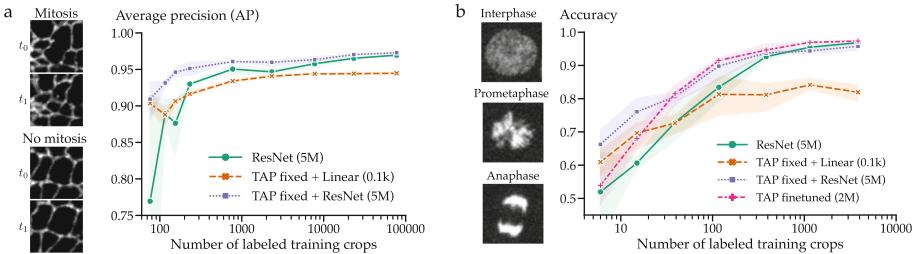


Fig. 4. a) Mitosis classification on FLYWING for two consecutive timepoints with TAP representations *vs.* a supervised ResNet baseline (green). **b)** Cell state classification in MDCK with fixed/fine-tuned TAP representations *vs.* a supervised ResNet baseline (green). We show results of three runs per model, # of params in parenthesis.

Mitosis Classification on Flywing: Since TAP attribution maps strongly highlight cell divisions, we consider predicting mitotic events an appropriate first downstream task to evaluate TAP. To that end, we generate a dataset of 97k crops of size $2 \times 96 \times 96$ from FLYWING and label them as mitotic/non-mitotic (16k/81k) based on available tracking data [20]. We train TAP networks on FLYWING and use a small ResNet architecture ($\approx 5M$ params) that is trained from scratch as a supervised baseline. In Fig. 4a we show average precision (AP) on a held-out test set while varying the amount of available training data. As expected, the performance of the supervised baseline drops substantially for low amounts of training data and surprisingly is already outperformed by a linear classifier (100 params) on top of TAP representations (*e.g.* 0.90 *vs.* 0.77 for 76 labeled crops). Training a small ResNet on fixed TAP representations consistently outperforms the supervised baseline even if hundreds of annotated cell divisions are available for training (*e.g.* 0.96 *vs.* 0.95 for 2328 labeled crops with ~ 400 cell divisions), confirming the value of TAP representations to detect mitotic events.

Cell State Classification on Mdck: Next we turn to the more challenging task of distinguishing between cells in interphase, prometaphase and anaphase from MDCK. This dataset consists of 4800 crops of size 80×80 that are labeled with one of the three classes (1600 crops/class). Again we use a ResNet as

supervised baseline and report in Fig. 4b test classification accuracy for varying amount of training data. As before, both a linear classifier as well as a ResNet trained on fixed TAP representations outperform the baseline especially in the low data regime, with the latter showing better or comparable results across the whole data regime (*e.g.* 0.90 vs. 0.83 for 117 annotated cells). Additionally, we finetune the pretrained TAP feature extractor for this downstream task, which slightly improves the results given enough training data. Notably, already at 30% training data it reaches the same performance (0.97) as the baseline model trained on the full training set.

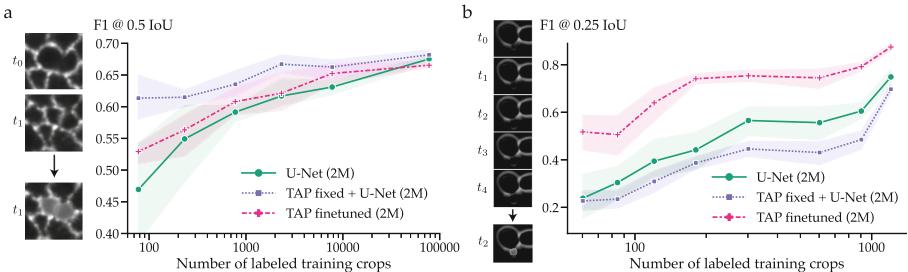


Fig. 5. **a)** Mitosis segmentation in FLYWING for two consecutive timepoints with fixed/finetuned TAP representations *vs.* a supervised U-NET baseline (green). We report F1 @ 0.5 IoU after removing objects smaller than 64 pixels. **b)** Emerging bud detection in YEAST from five consecutive timepoints with fixed/finetuned TAP representations versus a supervised U-NET baseline (green). We report F1 @ 0.25 IoU on 2D+time objects. We show results of three runs per model, # of params in parenthesis.

Mitosis Segmentation on Flywing: We now apply TAP on a pixel-level downstream task to fully exploit that the learned TAP representations are dense. We use the same dataset as for FLYWING mitosis classification, but now densely label post-mitotic cells. We predict a pixel-wise probability map, threshold it at 0.5 and extract connected components as objects. To evaluate performance, we match a predicted/ground truth object if their intersection over union (IoU) is greater than 0.5, and report the F1 score after matching. The baseline model is a U-NET trained from scratch. Training a U-NET on fixed TAP representations always outperforms the baseline, and when only using 3% of the training data it reaches similar performance as the baseline trained on all available labels (0.67 vs. 0.68, Fig. 5a). Interestingly, fine-tuning TAP only slightly outperforms the supervised baseline for this task even for moderate amounts of training data, suggesting that fixed TAP representations generalize better for limited-size datasets.

Emerging Bud Detection on Yeast: Finally, we test TAP on the challenging task of segmenting emerging buds in phase contrast images of yeast colonies. We train TAP networks on YEAST and generate a dataset of 1205 crops of size $5 \times 192 \times 192$ where we densely label yeast buds in the central frame (defined

as buds that appeared less than 13 frames ago) based on available segmentation data [17]. We evaluate all methods on held out test videos by interpreting the resulting 2D+time segmentations as 3D objects and computing the F1 score using an IoU threshold of 0.25. The baseline model is again a U-NET trained from scratch. Surprisingly, training with fixed TAP representations performs slightly worse than the baseline for this dataset (Fig. 5b), possibly due to cell density differences between TAP training and test videos. However, fine-tuning TAP features outperforms the baseline by a large margin (*e.g.* 0.64 *vs.* 0.39 for 120 frames) across the full training data regime, yielding already with 15% labels the same F1 score as the baseline using all labels.

4 Discussion

We have presented TAP, a self-supervised pretraining scheme that learns biologically meaningful representations from live-cell microscopy videos. We show that TAP uncovers sparse time-asymmetric biological processes and events in raw unlabeled recordings without any human supervision. Furthermore, we demonstrate on a variety of datasets that the learned features can substantially reduce the required amount of annotations for downstream tasks. Although in this work we focus on 2D+t image sequences, the principle of TAP should generalize to 3D+t datasets, for which dense ground truth creation is often prohibitively expensive and therefore the benefits of modern deep learning are not fully tapped into. We leave this to future work, together with the application of TAP to cell tracking algorithms, in which accurate mitosis detection is a crucial component.

Acknowledgements. We thank Albert Dominguez (EPFL) and Uwe Schmidt for helpful comments, Natalie Dye (PoL Dresden) and Franz Gruber for providing the FLYWING dataset, Benedikt Mairhörmann and Kurt Schmoller (Helmholtz Munich) for providing additional YEAST training data, and Alan Lowe (UCL) for providing the MDCK dataset. M.W. and B.G. are supported by the EPFL School of Life Sciences ELISIR program and CARIGEST SA.

References

1. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML, pp. 1597–1607 (2020)
2. Dorkenwald, M., Xiao, F., Brattoli, B., Tighe, J., Modolo, D.: SCVRL: shuffled contrastive video representation learning. In: CVPR, pp. 4132–4141 (2022)
3. Ericsson, L., Gouk, H., Loy, C.C., Hospedales, T.M.: Self-supervised representation learning: introduction, advances, and challenges. IEEE Sig. Process. Mag. **39**(3), 42–62 (2022)
4. Etournay, R., Popović, M., Merkel, M., Nandi, A., Blasse, C., Aigouy, B., et al.: Interplay of cell dynamics and epithelial tension during morphogenesis of the *Drosophila* pupal wing. eLife **4**, e07090 (2015)
5. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: ICLR, OpenReview.net (2018)

6. Greenwald, N.F., Miller, G., Moen, E., Kong, A., Kagel, A., et al.: Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning. *Nat. Biotechnol.* **40**(4), 555–565 (2021)
7. Han, H., Dmitrieva, M., Sauer, A., Tam, K.H., Rittscher, J.: Self-supervised voxel-level representation redisCOVERS subcellular structures in volume electron microscopy. In: CVPRW, pp. 1874–1883 (2022)
8. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: CVPR, pp. 16000–16009 (2022)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
10. Hsu, J., Gu, J., Wu, G., Chiu, W., Yeung, S.: Capturing implicit hierarchical structure in 3d biomedical images with self-supervised hyperbolic representations. In: NeurIPS, vol. 34, pp. 5112–5123 (2021)
11. Hu, K., Shao, J., Liu, Y., Raj, B., Savvides, M., Shen, Z.: Contrast and order representations for video self-supervised learning. In: ICCV, pp. 7939–7949 (2021)
12. Hua, T., Wang, W., Xue, Z., Ren, S., Wang, Y., Zhao, H.: On feature decorrelation in self-supervised learning. In: CVPR, pp. 9598–9608 (2021)
13. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (2015)
14. Lee, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Unsupervised representation learning by sorting sequences. In: ICCV, pp. 667–676 (2017)
15. Misra, I., Zitnick, C.L., Hebert, M.: Shuffle and learn: unsupervised learning using temporal order verification. In: ECCV, pp. 527–544 (2016)
16. Padovani, F., Mairhörmann, B., Falter-Braun, P., Lengefeld, J., Schmoller, K.M.: Segmentation, tracking and cell cycle analysis of live-cell imaging data with Cell-ACDC. *BMC Biol.* **20**, 174 (2022)
17. Padovani, F., Mairhörmann, B., Lengefeld, J., Falter-Braun, P., Schmoller, K.: Cell-ACDC: segmentation, tracking, annotation and quantification of microscopy imaging data (dataset). <https://zenodo.org/record/6795124> (2022)
18. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: feature learning by inpainting. In: CVPR, pp. 2536–2544 (2016)
19. Pickup, L.C., et al.: Seeing the arrow of time. In: CVPR, pp. 2043–2050 (2014)
20. Piscitello-Gómez, R., Gruber, F.S., Krishna, A., Duclut, C., Modes, C.D., et al.: Core PCP mutations affect short time mechanical properties but not tissue morphogenesis in the *Drosophila* pupal wing. *bioRxiv* (2022)
21. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
22. Schiappa, M.C., Rawat, Y.S., Shah, M.: Self-supervised learning for videos: a survey. *ACM Comput. Surv.* **55**(13s), 1–37 (2022)
23. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: visual explanations from deep networks via gradient-based localization. In: ICCV, pp. 618–626 (2017)
24. Stelzer, E.H.K., et al.: Light sheet fluorescence microscopy. *Nat. Rev. Methods Primers* **1**(1), 1–25 (2021)
25. Stringer, C., Wang, T., Michaelos, M., Pachitariu, M.: Cellpose: a generalist algorithm for cellular segmentation. *Nat. Methods* **18**(1), 100–106 (2021)
26. Tomer, R., Khairy, K., Keller, P.J.: Shedding light on the system: studying embryonic development with light sheet microscopy. *Curr. Opin. Genet. Dev.* **21**(5), 558–565 (2011)

27. Ulicna, K., Vallardi, G., Charras, G., Lowe, A.: Mdck cell tracking reference dataset. https://rdr.ucl.ac.uk/articles/dataset/Cell_tracking_reference_dataset/16595978
28. Ulicna, K., Vallardi, G., Charras, G., Lowe, A.R.: Automated deep lineage tree analysis using a bayesian single cell tracking approach. *Front. Comput. Sci.* **3**, 734559 (2021)
29. Ulman, V., Maška, M., Magnusson, K.E.G., Ronneberger, O., Haubold, C., et al.: An objective comparison of cell-tracking algorithms. *Nat. Methods* **14**(12), 1141–1152 (2017). <https://doi.org/10.1038/nmeth.4473>
30. Wei, D., Lim, J., Zisserman, A., Freeman, W.T.: Learning and using the arrow of time. In: CVPR, pp. 8052–8060 (2018)
31. Weigert, M., Schmidt, U., Haase, R., Sugawara, K., Myers, G.: Star-convex polyhedra for 3d object detection and segmentation in microscopy. In: WACV, pp. 3666–3673 (2020)
32. Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R.R., Smola, A.J.: Deep sets. In: NeurIPS (2017)
33. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: self-supervised learning via redundancy reduction. In: ICML, pp. 12310–12320 (2021)