Contrastive Learning Through Time

Anonymous Author(s)
Affiliation
Address
email

Abstract

Contrastive learning has emerged as a powerful form of unsupervised representation learning for images. The utility of learned representations for downstream tasks depends strongly on the chosen augmentation operations. Taking inspiration from biology, we here study contrastive learning through time (CLTT), that works completely without any augmentation operations. Instead, positive pairs of images are generated from temporally close video frames during extended naturalistic interaction with objects. We propose a family of CLTT algorithms based on state-of-the-art contrastive learning methods and test them on three data sets. We demonstrate that CLTT allows linear classification performance that approaches that of the fully supervised setting. We also consider temporal structure resulting from one object being seen systematically before or after another object. We show that this leads to increased representational similarity between these objects ("close in time, will align"), matching classic biological findings. The data sets and code for this paper can be downloaded at: (to be added for final version).

1 Introduction

A hallmark of biological organisms is their ability to learn to understand the world around them in a largely autonomous fashion. Consider learning about visual objects. A human infant is not exposed to object views sampled i.i.d. from some fixed distribution and conveniently labeled image by image. Instead, she forms representations of objects and categories during extended interactions with individual objects (Fig. 1A, Bambach et al. (2018)) and requires hardly any (verbal) labels for this (LaTourrette & Waxman, 2019). Mimicking such learning abilities in artificial systems would represent a giant leap forward for artificial intelligence.

Self-supervised learning has emerged as a promising alternative to fully supervised approaches. In the domain of visual object recognition, recent contrastive learning approaches have obtained strong results on standard object recognition benchmarks (Chen et al., 2020a,b; Mitrovic et al., 2021; Grill et al., 2020). These approaches rely on a range of so-called augmentation operations. The basic idea is that an image is transformed through a number of operations (e.g., scaling, flipping, cropping, rotating, blurring, color distortions, pixel noise, ...) that change its appearance but not its meaning (e.g., "cat"). The key mechanism of contrastive learning is to form a representation where such augmented versions of an image are mapped on to close-by latent representations, while at the same time avoiding a "collapse" of the representation, i.e., making sure the network does not simply map all inputs to the same point in the latent space. Not surprisingly, the quality of the learned representations for downstream tasks strongly depends on the chosen augmentation operations (Grill et al., 2020).

Biological evidence suggests that the temporal structure of visual inputs is essential for shaping object representations. Miyashita (1988) showed images of fractal-like objects to monkeys in always the same order. As the monkeys learned to represent these images in their visual cortices, Miyashita observed that the representations of objects which were neighbors in the sequence became aligned

Submitted to 3rd Workshop on Shared Visual Representations in Human and Machine Intelligence (SVRHM 2021) of the Neural Information Processing Systems (NeurIPS) conference.

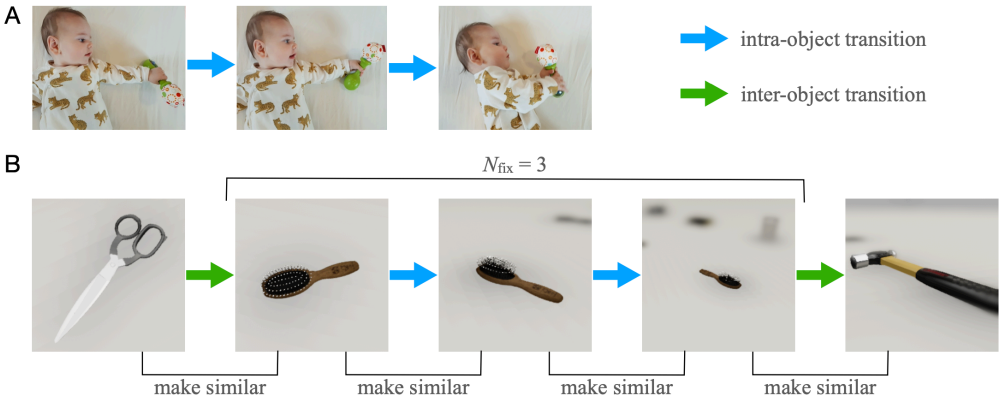

Figure 1: Contrastive Learning Through Time (CLTT). **A.** Infants learn about objects during extended interactions, where they experience different views of an object before directing their attention elsewhere. **B.** Our CLTT approach mimics the essence of such interactions. A certain number N_{fix} of object views are sampled before directing attention to another object. Latent representations of successive views (both intra-object and inter-object transitions) are made more similar.

— even if these objects were visually dissimilar. More generally, by learning a representation such that subsequent views are mapped onto close-by latent codes (Fig. 1B), a representation may emerge that maps different views of the same object onto similar latent codes, thereby establishing (partial) *invariance*. Indeed, a long history of neural network and machine learning models have exploited temporal structure for unsupervised representation learning in this way (Földiák, 1991; Rolls & Milward, 2000; Wiskott & Sejnowski, 2002). However, this idea has not been systematically explored in a contrastive learning context. We call this approach *contrastive learning through time* (CLTT). The basic idea is to replace the augmentation operations in contrastive learning with natural appearance variation occurring during extended object interactions.

A related approach has recently been proposed by Orhan et al. (2020). They consider learning on a longitudinal headcam video set from three developing children (Sullivan et al., 2020). They focus on a *temporal classification* approach, where they divide the videos into a finite number of contiguous segments of the same length that they call *temporal classes*. The learning objective is to predict from which of the classes a particular video frame originates. They also consider a temporal contrastive learning objective with the MoCo contrastive learning implementation of Chen et al. (2020b). This objective aims to make the latent representations of adjacent video frames similar and can therefore be considered a specific instance of CLTT. Another related approach is that of Knights et al. (2021), who learn embeddings of video clips. Their learning objective makes latent codes of adjacent frames within a video clip similar, while making them distinct from latent codes of frames from other video clips. Note that this setup requires the video clips to be segmented, i.e., the system has access to the information where each video starts and ends rather than being exposed to an unlabeled continuous video stream as in Orhan et al. (2020) and CLTT.

To systematically study CLTT, we propose a family of new algorithms and test them on new and existing data sets. We also simulate the classic experiments by Miyashita (1988) demonstrating that objects form similar latent representations in the brain when they are systematically seen one after the other, even if they are visually dissimilar. We summarize our contributions as follows:

- We develop a general CLTT framework and introduce a family of CLTT algorithms based on state-of-the-art contrastive learning methods.
- We introduce novel data sets to study CLTT under controlled conditions.
- We systematically analyze the conditions for CLTT to be successful and demonstrate that it can approach fully supervised methods.
- We show that CLTT maps objects that are systematically seen in temporal succession on similar latent representations, matching classic neurobiological findings.

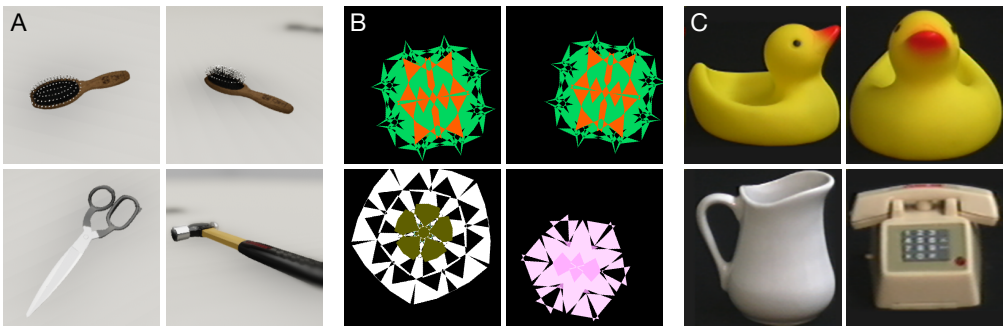

Figure 2: Data sets developed/used in our study. **A.** TDW data set containing common household objects viewed from different orientations and distances. The upper two images show the same example object ("hair brush") from two perspectives. The lower two images show two other example objects **B.** Fractal-like objects inspired by Miyashita (1988). **C.** Example objects from COIL-100.

2 Methods Overview

The general idea of CLTT is to use successive input images as positive pairs in contrastive learning. CLTT can therefore be combined with a wide range of contrastive learning algorithms. For our experiments we have chosen three state-of-the-art contrastive learning methods: SimCLR (Chen et al., 2020a), RELIC (Mitrovic et al., 2021) and BYOL (Grill et al., 2020). We refer to the CLTT versions of these algorithms as SimCLR-TT, RELIC-TT, and BYOL-TT, respectively. In all cases, we use a ResNet-18 architecture (He et al., 2016) followed by a linear layer to transform the input images into a 128-dimensional latent representation. Algorithm details are given in Appendix A.1.

We test these algorithms with three data sets (Fig. 2). We constructed a new data set using ThreeD-World (Gan et al., 2021). It consists of 12 household objects viewed from a total of 1008 different perspectives and distances. To model the biological experiments of Miyashita (Miyashita, 1988), we constructed a new fractal object dataset. Finally, we use the COIL-100 data set (Nene et al., 1996) to test CLTT on real images taken with a camera. Additional details are given in Appendix A.2.

To generate sequences of object views for CLTT, we sample different kinds of paths through the space of possible views for each data set. Our sampling procedure is a simple model of how infants explore the world and interact with objects, where gaze typically stays on an object for multiple fixations before being redirected elsewhere. Therefore, a sequence of images showing the same object from different view points is followed by a switch to another object. We denote the number of fixations on a single object before moving to the next object N_{fix}. We consider two procedures for sampling successive views of the same object. In the *random walk* procedure, successive views are generated by changing the viewing direction by one step along the azimuth, elevation or distance dimension (depending on the data set used). In the *uniform* approach, we sample successive views uniformly at random across all dimensions. This latter approach does not attempt to model infant looking sequences, but it allows us to test the effect of the presence or absence of correlations across successive views. For gaze shifts to a different object we also consider two different procedures. In the *random object sequence* method, the next object is chosen uniformly at random while excluding the current object. In the *fixed object sequence* procedure, the objects are always viewed in the same order (as in the experiments by Miyashita (1988)).

3 Results

3.1 CLTT approaches representation quality resulting from supervised learning

We evaluate the proposed family of CLTT methods, namely SimCLR-TT, RELIC-TT, BYOL-TT on the novel ThreeDWorld data set (Fig. 3). We also add a supervised method as baseline, which has access to the true label of each image. We focus on the *random walk* sampling of views and the *random object sequence* procedure for sampling objects. We train the network for 100 epochs and vary N_{fix} to study its effect. In order to evaluate the quality of learned representations, we use

a Linear Least Squares (LLS) classifier to test linear separability. We also quantify the quality of clustering in the latent space by calculating the average Within Cluster Sum of Squares (WCSS) divided by the average Between Clusters Sum of Squares (BCSS) (MacQueen et al., 1967). In general, a good representation, like that resulting from supervised learning, should be linearly separable and have high LLS classification accuracy. Similarly, a low value of the WCSS to BCSS ratio indicates strong clustering. In Fig. 3A we show the LLS classification accuracy for $N_{\text{fix}} = 30$ as a function of training epoch. The RELIC-TT method performs best in this setting and approaches the quality of the supervised learning. Figure 3B shows the final LLS classification accuracy after 100 training epochs for different values of N_{fix}. Interestingly, while for RELIC-TT and SimCLR-TT greater N_{fix} improves performance, this is not the case for BYOL-TT. Figure 3C depicts a PacMAP (Wang et al., 2020) visualization of the latent space resulting from training with RELIC-TT for 100 epochs using $N_{\text{fix}} = 30$. Results for the WCSS to BCSS ratio roughly mirror the LLS results and are shown in Appendix A.3. We also show experiments on the classic COIL-100 data set (Nene et al., 1996) to test the approach with real (non computer-rendered) images in Appendix A.4.

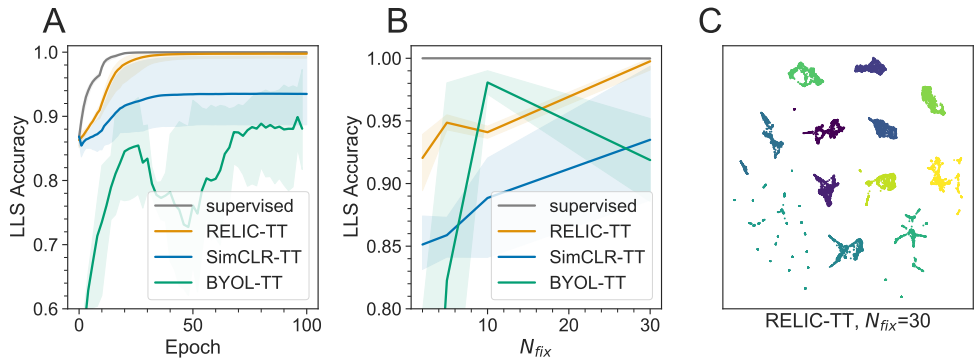

Figure 3: Results for CLTT on the TDW data set. **A.** Comparison of Linear Least Squares (LLS) classification accuracy as a function of training epoch for the different algorithms and with $N_{\text{fix}} = 30$. **B.** Final LLS classification accuracy as a function of N_{fix}. **C.** Visualization of the clustering of representations in the latent space using PacMAP, here shown for RELIC-TT with $N_{\text{fix}} = 30$. Each color corresponds to one distinct object. Shaded areas in panels A and B represent the standard deviation based on three individual runs.

3.2 CLTT aligns latent codes of successively viewed objects

To relate our framework to biological findings we use the Miyashita-style data set and train with our SimCLR-TT approach using the *fixed object sequence* procedure. Due to the unique properties of this data we used a slightly modified procedure for sampling positive pairs (see App. A.2). We train the network for 10^6 stimulus presentations (up to 10^3 epochs) with batchsize 100 and vary N_{fix}. In line with results from neuroscience, fractals that were presented in succession evoke more similar activations in the latent space than fractals that are far apart in the predefined sequence. Figure 4A depicts the mean cosine similarity between a fractal's latent representation and that of its 2 n-th nearest neighbors along the sequence (in the positive and negative time direction). Note that neighbor zero has a similarity of one, as it is de-facto the same activation pattern. The curves for all values of N_{fix} display a downward slope that is clearly separated from the baseline (dashed lines). This indicates that over time not only immediate neighbors, but also more distant fractals become associated. In fact for $N_{\text{fix}} = 2$ we see significant deviations for the three next neighbors before approaching baseline ($p < .001$, two-sided Kolmogorov-Smirnov two-sample test). This experiment illustrates a fundamental property of CLTT that can be directly related to biological findings. We also compare our results to supervised learning, which can solve the classification of 100 fractals, but the resulting latent representations do not display any structure of temporal associations (not shown).

We also test if the alignment of latent representations of objects that are consistently shown in succession is also be observed with the TDW data set. For this we sample objects using the *fixed object sequence* procedure, i.e., the objects were always seen in the same sequence order. We focus

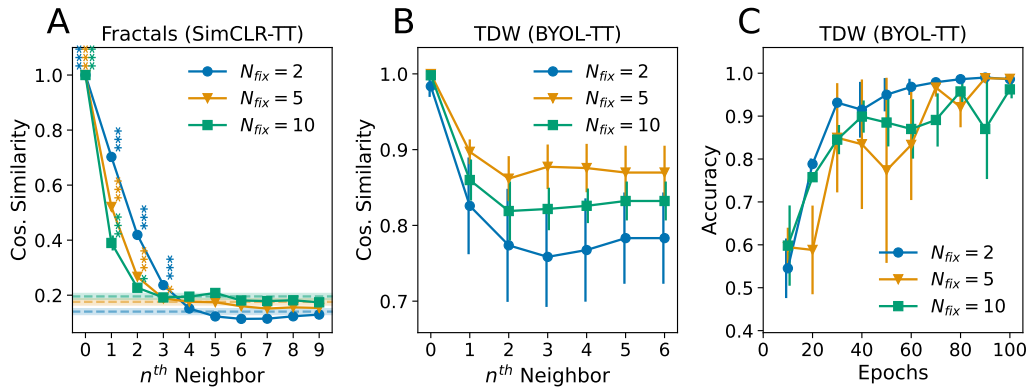

Figure 4: Latent codes of successively viewed objects. **A.** Cosine similarity of neighboring fractal objects along the timeline as seen in the latent space for different values of N_{fix}. Colored dashed lines and envelopes represent the mean and standard deviation of the neighbors not shown, i.e., $n \geq 10$. Error-bars depict the standard deviation based on three independent runs. **B.** Same for TDW objects. **C.** Test set accuracy for TDW data after training a linear readout on top of the ResNet trained with BYOL-TT for different values of N_{fix}.

on BYOL-TT for this experiment. Figure 4B replicates the effect from the Fractals data set (compare Fig. 4A). Figure 4C shows the LLS classification accuracy under these conditions. The fixed sequence order aids classification and BYOL-TT achieves excellent classification performance for different values of N_{fix}. Thus, the network is able to learn good representations of the input data even with low values of N_{fix} when the order in which the objects are shown to the network is fixed.

4 Discussion

We have developed a general framework for contrastive learning through time (CLTT). CLTT replaces conventional augmentation operations used in contrastive learning with the natural variability occurring during temporally extended object interactions such that subsequent inputs occurring during such interactions are mapped onto close-by latent representations ("close in time, will align"). To systematically investigate this approach, we have created a new data set using the ThreeDWorld environment (Gan et al., 2021) and a new fractal-like data set inspired by biological experiments and we have also used the classic COIL-100 computer vision data set. We have demonstrated that CLTT can approach the quality of representations learned using full supervision and that it is able to reproduce classic biological findings showing that inputs that occur close in time are mapped onto close-by latent representations by the brain.

Our work has a number of limitations. First, while the use of TDW gives us perfect control over all parameters of viewing sequences including scene geometry, sequence of views, lighting conditions, etc., the approach needs to be validated in the real world. Using the COIL-100 data set has been a first step in this direction. To address this more thoroughly, future work could consider first person videos from infants wearing head-mounted cameras (Bambach et al., 2018; Orhan et al., 2020). Second, we have assumed well-separated objects without major occlusions in front of uniform backgrounds. Learning in a cluttered environment is expected to be harder, but it may benefit from foveated vision and additional attentional and figure-ground separation mechanisms, which we plan to incorporate in future work. Third, our eye movement model, which kept gaze on the same object for a certain number of fixations before redirecting it elsewhere, is overly simplistic. In the future, it will be interesting to use more refined models that better reflect (measured) gaze sequences of children and adults. Indeed, it is an interesting question what gaze sequences are particularly beneficial for learning and if and how infants and artificial vision systems can optimize viewing sequences to maximize their own learning progress. This links CLTT to research on intrinsic motivation and (artificial) curiosity. Exploring these issues will bring us closer to building artificial vision systems that can learn truly autonomously.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We mention limitations in the Discussion.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We will provide links to our code and data sets in the final version.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Note that some details are given in the Appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Experiments were repeated with different random seeds and error bars are given.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Note that some details are given in the Appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite the publications on the ThreeDWorld software and the COIL-100 data set.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] A link to the code for the fractal generation will be
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

References

Sven Bambach, David Crandall, Linda Smith, and Chen Yu. Toddler-inspired visual object learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/48ab2f9b45957ab574cf005eb8a76760-Paper.pdf.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020a.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Peter Földiák. Learning invariance from transformation sequences. *Neural computation*, 3(2): 194–200, 1991. doi: 10.1162/neco.1991.3.2.194.

Chuang Gan, Jeremy Schwartz, Seth Alter, Damian Mrowca, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, Megumi Sano, Kuno Kim, Elias Wang, Michael Lingelbach, Aidan Curtis, Kevin Tyler Feigelis, Daniel Bear, Dan Gutfreund, David Daniel Cox, Antonio Torralba, James J. DiCarlo, Joshua B. Tenenbaum, Josh Mcdermott, and Daniel LK Yamins. ThreeDWorld: A platform for interactive multi-modal physical simulation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. URL https://openreview.net/forum?id=db1InWAwW2T.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 21271–21284. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Joshua Knights, Ben Harwood, Daniel Ward, Anthony Vanderkop, Olivia Mackenzie-Ross, and Peyman Moghadam. Temporally coherent embeddings for self-supervised video representation learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 8914–8921. IEEE, 2021.

Alexander LaTourrette and Sandra R Waxman. A little labeling goes a long way: Semi-supervised learning in infancy. *Developmental science*, 22(1):e12736, 2019. doi: 10.1111/desc.12736.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

James MacQueen et al. Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 1(14): 281–297, 1967.

Jovana Mitrovic, Brian McWilliams, Jacob C Walker, Lars Holger Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=9p2ekP904Rs.

Yasushi Miyashita. Neuronal correlate of visual associative long-term memory in the primate temporal cortex. *Nature*, 335(6193):817–820, 1988. doi: 10.1038/335817a0.

Yasushi Miyashita, Sei-Ichi Higuchi, Kuniyoshi Sakai, and Naohiko Masui. Generation of fractal patterns for probing the visual memory. *Neuroscience Research*, 12(1):307–311, 1991. doi: 10.1016/0168-0102(91)90121-E.

Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-100). Technical report, Columbia University, Feb 1996.

Emin Orhan, Vaibhav Gupta, and Brenden M Lake. Self-supervised learning through the eyes of a child. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9960–9971. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/7183145a2a3e0ce2b68cd3735186b1d5-Paper.pdf`.

Edmund T Rolls and T Milward. A model of invariant object recognition in the visual system: learning rules, activation functions, lateral inhibition, and information-based performance measures. *Neural computation*, 12(11):2547–2572, 2000. doi: 10.1162/089976600300014845.

Jessica Sullivan, Michelle Mei, Andrew Perfors, Erica Wojcik, and Michael C Frank. Saycam: A large, longitudinal audiovisual dataset recorded from the infant's perspective. *Open Mind*, pp. 1–10, 2020. doi: 10.1162/opmi_a_00039.

Yingfan Wang, Haiyang Huang, Cynthia Rudin, and Yaron Shaposhnik. Understanding how dimension reduction tools work: an empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization. *arXiv preprint arXiv:2012.04456*, 2020.

Laurenz Wiskott and Terrence J Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770, 2002. doi: 10.1162/089976602317318938.

A Appendix

A.1 Details on the Algorithms

A.1.1 SimCLR-TT

SimCLR (Chen et al., 2020a) is an effective contrastive learning approach for visual representation learning. By defining a wide range of augmentation operations and treating differently augmented images as positive samples, SimCLR has achieved state-of-the-art performance. Its groundbreaking success has triggered a surge of interest in augmentation-based self-supervised learning.

In the original SimCLR method, every sample has two differently augmented views. For one sample, these two views form a positive pair, and each of the two views forms negative pairs with all the other views of other samples in a batch. Incorporating the notion of CLTT that successively shown images should have similar representations, we define SimCLR-TT to use subsequent images to form positive pairs. Specifically, we divide the sequence of views into windows where the object identity stays the same. A training batch consists of several such windows. To every window we add one "mixing view" from a randomly chosen other object. Then a view in a particular window will form positive pairs with all the views in the same window (including the mixing view of the other object) and form negative pairs with all the other views in other windows. The loss of SimCLR-TT is then defined as follows:

$$\mathcal{L}\left(z_i\right) = -\log \frac{\sum_{j \in W(i),\, j \neq i} \exp\left(\operatorname{sim}\left(z_i, z_j\right)\right)}{\sum_{k \notin W(i)} \exp\left(\operatorname{sim}\left(z_i, z_k\right)\right)}, \tag{1}$$

where z_i is the latent code of frame i and we denote the window to which image i belongs as $W\left(i\right)$. We use a Gaussian Radial Basis Function as the similarity function $\operatorname{sim}\left(x, y\right) = \exp\left(-\left\|x - y\right\|\right)$. We omitted the temperature parameter τ in the original SimCLR loss as well as the shape parameter ε in the RBF function for simplicity. In our SimCLR-TT implementation, we use the AdamW optimizer (Loshchilov & Hutter, 2019) with a starting learning rate of 10^{-3}, which decays by a factor of 0.3 after every 10 epochs. The batch size for training depends on $N_{\mathbf{fix}}$ and should be slightly above 256 to make sure that there is about the same number of windows for each object. We train the model with a single GeForce Titan X GPU with 12 GB graphical memory. One training run for 100 epochs takes approximately 25 minutes to complete on the ThreeDWorld data set.

A.1.2 RELIC-TT

RELIC (Mitrovic et al., 2021) is an approach that uses an additional penalty loss compared to SimCLR. It obtains state-of-the-art results by keeping the similarity distribution of one sample invariant against differently augmented views of other samples. Incorporating this notion into the CLTT approach, we derive RELIC-TT by adding another loss term that minimizes the standard deviation of the similarities of one sample against other samples within their sliding windows. The loss can be described as follows:

$$\mathcal{L}_p\left(x_i\right) = \sum_{k=1}^{N} \mathrm{std}_{j \in W_k}\left(\mathrm{sim}\left(z_i, z_j\right)\right), \tag{2}$$

where we minimize the standard deviation of similarities between the anchor x_i with other samples within their fixation window W_k. The training configuration and hardware are the same as for SimCLR-TT.

A.1.3 BYOL-TT

BYOL-TT builds on the Bootstrap Your Own Latent (BYOL) architecture by (Grill et al., 2020). BYOL has been shown to outperform other contrastive learning architectures like SimCLR (Chen et al., 2020a) or MoCov2 (Chen et al., 2020b) on the ImageNet data set (Deng et al., 2009). A key advantage of the BYOL architecture is that it works without negative pairs, which sets it apart from other contrastive learning algorithms. Instead, it uses a second so-called target network. The target network receives an augmented version of the input data like the online network. It will produce a target projection and the online network tries to predict this target projection. The loss function minimizes the similarity between the target projection and the prediction of the online network.

In our BYOL-TT implementation, we use the ADAM optimizer with a learning rate of 2×10^{-4} and a batch size of 256. We trained the model with a Nvidia GeForce Titan X GPU with 12 GB memory. One training run with 100 epochs takes approximately 1 hour to complete. The loss is given by:

$$\mathcal{L}_{\theta,\xi} = 2 - 2 \cdot \frac{\langle q_\theta(z_\theta), z'_\xi \rangle}{||q_\theta(z_\theta)||_2 \cdot ||z'_\xi||_2}, \tag{3}$$

where $q_\theta(z_\theta)$ is the prediction of the online network of z'_ξ. θ corresponds to the weights of the online network and ξ represent the weights of the target network. The optimization will be performed with respect to θ, the weights ξ of the target network will be updated using an exponential moving average of the online network. Here we choose $\tau = 0.99$ as target decay rate. In the original BYOL architecture an input image x produces two augmented views v and v' which are then shown to target and online network. In our case no augmentations are applied to the input image, positive pairs are formed by images next to each other in the presented data timeline. Two consecutive images in this timeline will then be used as views v and v'.

A.1.4 Intra-class vs. Inter-class positive pairs in CLTT

Positive pairs are formed with a sliding window approach — images next to each other in a viewing sequence form a positive pair during contrastive learning. The sampling procedures introduced in the main text give rise to two types of positive pairs: *intra-class pairs* show different views of the same object and *inter-class pairs* show different objects. The fraction of intra-class pairs is given by $(N_{\text{fix}} - 1)/N_{\text{fix}}$, while the fraction of inter-class pairs is $1/N_{\text{fix}}$). In other words, there are $N_{\text{fix}} - 1$ times as many intra-class pairs than inter-class pairs. In the *random sequence* method, all possible $N_{\text{obj}}(N_{\text{obj}} - 1)/2$ inter-class pairs occur, whereas in the *fixed sequence* procedure only N_{obj} distinct types of inter-class pairs are present.

A.2 Additional Details on the Data Sets

A.2.1 ThreeDWorld data set

This data set was created with the ThreeDWorld (TDW) software (Gan et al., 2021), which is published under the BSD 2-Clause "Simplified" License. TDW was chosen, because it enables near-photorealistic rendering including depth-of-field effects and the possibility of simulating physical interaction with objects (which we plan to utilize in the future). Our data set comprises twelve high

quality 3D models of common household objects such as a hair brush, a hammer or scissors (compare Fig. 2A). They are arranged in a 4×3 grid on a white floor. To simulate viewing sequences that an infant may experience while interacting with an object such as turning an object in hand or moving relative to an object while fixating it, we render object views from different directions and distances. The individual images of the data set can then be arranged into videos simulating different viewing sequences, where an object is seen from different directions and distances. Specifically, to create the different views, we define a spherical coordinate system around the center of each object. Different views are created by changing viewing direction in terms of azimuth (from $0°$ to $350°$ in $10°$ steps), elevation (from $10°$ to $70°$ in $10°$ steps), and distance (0.3 cm to 0.6 cm in 10 cm steps). This gives rise to 1008 views per object and a total of 12,096 images, which are down-sampled to the size of 64 by 64 pixels. Note that due to the varying distances from the objects, sometimes not all parts of an object are visible inside the image and sometimes (parts of) other objects may be present. These often appear blurry due to the simulated depth-of-field effect.

A.2.2 Miyashita-style data set

The Miyashita-style data set draws inspiration from the neuroscience experiments from Miyashita (1988). We adhered to the generation procedure described in (Miyashita et al., 1991) and wrote python code to generate 100 different fractals of size 64×64. Each of those fractals is unique and highly distinguishable from all others. For our experiments, we give each fractal its own class label and allow for a certain degree of variability to simulate fixation inaccuracies during viewing of the images. Each fixation applies a transformation with a random rotation between -10 and $+10$ degrees, a random rescaling between 90 and $100\ \%$ and a random translation of up to $15\ \%$ in x- and y-direction, compare Fig. 2B. The fractals are used exclusively with the *fixed sequence* procedure, i.e., there is a predefined order of fractals that is repeatedly shown to the network.

Due to its small size of 100 distinct images and the comparably high number of classes (100 classes) we modify the sampling procedure of SimCLR-TT to accommodate these limitations. We first construct the entire time-sequence of fixations depending on the value of N_{fix}. We then utilize this time-sequence as a buffer to fill the batches with randomly sampled images (without replacement). This ensures the necessary diversity within a batch without increasing the batch size. For every image in the batch we sample a contrast that is one time step apart from the originally sampled image. This ensures having the same positive pairs as if one would go through the time-sequence in the original order.

As with the original SimCLR (Chen et al., 2020a) a batch size of 100 then yields 198 negative examples per positive pair from both contrasting views. This is unlike the approach mentioned above where one positively associates all images within a window and proves useful when committing to fixed batch-sizes and large values of N_{fix}.

We train the ResNet-18 network for 10^6 stimuli — or 10^5 batch-presentation swith the ADAM optimizer. We use a Nvidia GeForce RTX 2070 Super GPU with 8 GB memory. One training run takes approximately 2 hours on average to complete. For hyper-parameters we choose a learning rate of $\eta_{\max} = 0.5$ with cosine-annealing, a corresponding decay rate of 0.1 and a minimum learning rate of $\eta_{\min} = 5 \times 10^{-3}$. The temperature parameter was set to $\tau = 0.1$. For the experiments with the Miyashita-style data set the similarity function is chosen to be the cosine similarity, i.e., $\text{sim}(\mathbf{u}, \mathbf{v}) = \mathbf{u}^\top \mathbf{v} / \|\mathbf{u}\| \, \|\mathbf{v}\|$ and the original SimCLR loss was used.

A.2.3 COIL-100 data set

To validate the CLTT approach on real (rather then computer rendered) images, we employ the Columbia Object Image Library (COIL-100) database (Nene et al., 1996). It is composed of color images of 100 objects each viewed from 72 different directions by placing the objects on a motorized turntable. The objects are seen against a homogeneous black background (compare Fig. 2C). This data set is intended for non-commercial research purposes only. We use the ADAMW optimizer with a learning rate of 1×10^{-3} and a batch size of 256. For N_{fix} equals to 2, we train the model with one Nvidia GeForce GTX 1060 GPU with 6GB memory. One training run with 50 epochs takes approximately 22 minutes to complete. For N_{fix} equals to 5, 10, and 15, we train the model with a Nvidia GeForce Titan X GPU with 12GB memory. One training run with 50 epochs takes approximately 17 minutes to complete.

A.3 ThreeDWorld: WCSS/BCSS results

Figure A.1 shows the ratio between within class sum of squares (WCSS) and between class sum of squares (BCSS). This ratio gives a good indication of how well the clusters are separated. The different panels show the results for different values of N_{fix} for the different algorithms.

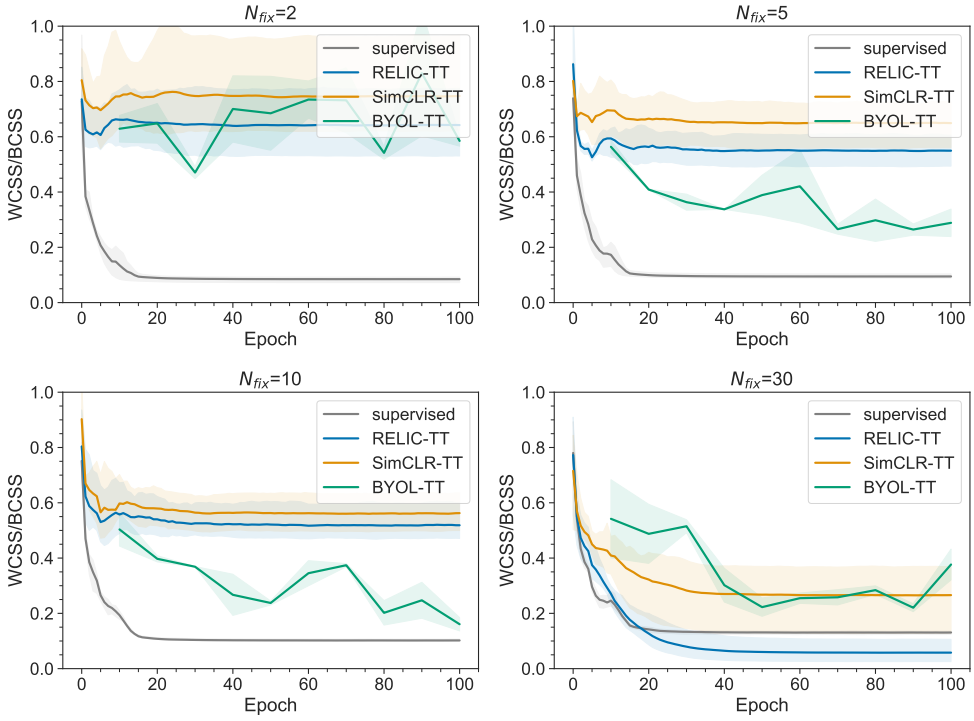

Figure A.1: WCSS/BCSS results of different methods for the ThreeDWorld data set.

A.4 Results with COIL-100 data set

We compare performance of SimCLR-TT, RELIC-TT, and the supervised method for different values of N_{fix}. We also compare the two sampling strategies: *random walk sampling*, where successive views are generated via a random walk in the space of viewing directions and *uniform sampling*, where successive views are picked uniformly at random across all possible viewing directions. Figure A.2A shows the LLS accuracy of the different algorithms and sampling strategies as a function of the number of training epochs for $N_{\text{fix}} = 5$. As before, the RELIC-TT method outperforms SimCLR-TT. Furthermore, we observe that the *uniform sampling* leads to better results than the *random walk* sampling. This is not surprising, as it creates more diversity in the training data by allowing for very different views of an object to be grouped as a positive pair during contrastive learning. Also, although learning is somewhat slower, the performance of RELIC-TT with uniform sampling approaches that of fully supervised learning. Figure A.2B compares the final LLS classification accuracy of the algorithms after 50 training epochs for different values of N_{fix}. For this data set an intermediate number of $N_{\text{fix}} = 5$ leads to the best performance. Figure A.2C visualizes the clustering of latent representations with PacMAP for RELIC-TT with $N_{\text{fix}} = 5$ and the *uniform sampling*. The 100 different objects have formed distinct clusters in the latent space.

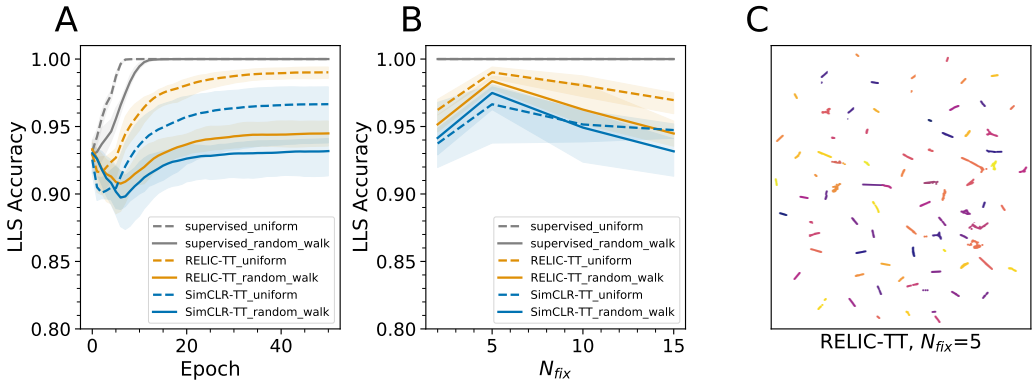

Figure A.2: Results for CLTT on the COIL-100 data set. **A.** Comparison of LLS classification accuracy for the different algorithms and sampling strategies with $N_{\text{fix}} = 5$. **B.** Influence of N_{fix} on LLS classification accuracy under the two sampling strategies. **C.** Visualization of the clustering of representations in the latent space using PacMAP, here shown for RELIC-TT with $N_{\text{fix}} = 5$ and the *uniform sampling* procedure. Different colors correspond to different objects, but due to the large number of objects, colors have been reused multiple times. Shaded areas in panels A and B represent the standard deviation based on five individual runs.