Abstract

Biological vision systems are unparalleled in their ability to learn visual representations without supervision. In machine learning, contrastive learning (CL) has led to major advances in forming object representations in an unsupervised fashion. These systems learn representations invariant to augmentation operations over images, like cropping or flipping. In contrast, biological vision systems exploit the temporal structure of the visual experience. This gives access to “augmentations” not commonly used in CL, like watching the same object from multiple viewpoints or against different backgrounds. Here, we systematically investigate and compare the potential benefits of such time-based augmentations for learning object categories. Our results show that time-based augmentations achieve large performance gains over state-of-the-art image augmentations. Specifically, our analyses reveal that: 1) 3-D object rotations drastically improve the learning of object categories; 2) viewing objects against changing backgrounds is vital for learning to discard background-related information. Overall, we conclude that time-based augmentations can greatly improve contrastive learning, narrowing the gap between artificial and biological vision systems.

1 Introduction

Learning object representations without supervision is a grand challenge for artificial vision systems. Recent approaches for visual contrastive learning (Grill et al., 2020; Chen et al., 2020) acquire representations invariant to data-augmentations based on image-based operators like crop/resize, blur or color distortion. The nature of these augmentations determines what information is retained and what information is discarded and therefore how useful these augmentations are for particular downstream tasks (Jaiswal et al., 2021; Tsai et al., 2020).

Biological vision systems, in contrast, exploit the temporal structure of visual input during interactions with objects for unsupervised representation learning. According to the slowness principle (Wiskott and Sejnowski, 2002; Li and DiCarlo, 2010; Wood and Wood, 2018), biological vision systems strive to discard high-frequency variations (e.g. individual pixel intensities) and retain slowly varying information (e.g. object identity) in their representation. Incorporating this idea into contrastive learning
learning approaches has led to a range of time-contrastive learning methods that learn to map inputs occurring close in time onto close-by latent representations (Oord et al., 2018; Schneider et al., 2021). Importantly, a human infant learning about objects will interact with these objects in manifold ways (Smith et al., 2018), which will determine what aspects of the visual input vary slowly vs. quickly. First, infants execute saccadic eye movements towards salient features and accommodate their pupils to the distance of the fixation point. This enables them to keep object features in focus while blurring background features due to their limited depth of field. Second, infants rotate, elevate, bring closer/farther objects while playing with them (Byrge et al., 2014). Third, as they gain mobility, they can move in the environment while holding an object, viewing it in different contexts and against different backgrounds. Here, we refer to such infant-inspired interactions as natural interactions.

We systematically study the impact of such natural interactions on representations learnt through time-contrastive learning. To simulate the interactions available to an infant, we investigate time-based augmentations in two settings. First, we introduce a new simulation environment based on the near-photorealistic simulation platform ThreeDWorld (TDW) (Gan et al., 2021) and combine it with a recent dataset of thousands of 3D object models (Toys4k) (Stojanov et al., 2021). We simulate an agent engaging in natural interactions while “playing” with objects in its home. Second, we dynamically re-segment real-world videos of object manipulations from the CORe50 dataset (Lomonaco and Maltoni, 2017) to approximate different types of natural interactions and systematically study the impact of relative frequencies of the interactions on the learned representation.

Our experiments show that adding time-based augmentations to conventional data-augmentations considerably improves category recognition. Our ablation study demonstrates that the benefit of time-based augmentations has two origins: 1) 3-D object rotations boost generalization across object shapes. 2) Viewing objects against different backgrounds while moving with them effectively reduces harmful effects of background clutter. Furthermore, we note that the combination of saccades and distance changes of objects have a similar effect to the standard crop/resize augmentation. We conclude that exploiting natural interactions via time-contrastive learning greatly improves unsupervised visual representation learning.

2 Related work

SimCLR-TT. SimCLR is a recent state-of-the-art, unsupervised, contrastive learning approach that manages to learn visual representations suitable to downstream tasks (Chen et al., 2020). The general idea is that two semantically close/different inputs should be close/distant in the learnt representation space. The underlying assumption is that applying certain transformations does not change the semantic meaning or label of an image, and thus should only slightly change the representation of that image. They sample an image $i$, apply transformations to it taken from a predefined set of transformations, resulting in two new images $x_i$ and $x_j$, also called a positive pair. The same procedure is applied to a batch of $N-1$ sampled different images resulting in $2N-2$ images, also called negatives samples. Its time-based variant, SimCLR-TT (Schneider et al., 2021; Oord et al., 2018), assumes that semantically close images are also close in time, such that $x_i$ and $x_j$ are successors in time. Then, both SimCLR and SimCLR-TT learn the parameters $\theta$ of the representation encoder (a neural network) $f_\theta$ to minimize:

$$\mathcal{L}_{i,j} = -\log \frac{e^{\text{sim}(f_\theta(x_i), f_\theta(x_j))/\tau}}{\sum_{k=1,k\neq i,k\neq j}^{2N} e^{\text{sim}(f_\theta(x_i), f_\theta(x_k))/\tau}},$$

where sim is a similarity function and $\tau$ is a temperature hyper-parameter. The numerator makes sure that positive pairs are close in the latent space. The denominator ensures that the image embeddings stay relatively distant from each other and do not collapse to a single point. The quality of the learned representation is typically assessed by training a linear readout on top of the learned representation in a supervised fashion.

Visual data-augmentations. Most self-supervised learning approaches use unnatural pretext tasks to learn representations. The most used ones are color distortion, cropping/resizing a part of an image, the horizontal flipping of an image, grayscaling the image and blurring the image (Chen et al., 2020). Such augmentations allow to extract important task-relevant features (Chen et al., 2020; Grill et al., 2020; Bardes et al., 2022). We show in our experiments that time-based augmentations outperform...
these standard ones and can be combined with them. Other augmentations can be categorized in three ways (Shorten and Khoshgoftaar, 2019; Jaiswal et al., 2021): 1) geometric augmentations include image rotations (Chen et al., 2020) or image translations (Shorten and Khoshgoftaar, 2019); 2) Context-based augmentations include jigsaw puzzle augmentations (Noroozi and Favaro, 2016; Misra and Maaten, 2020), pairing images (Inoue, 2018), greyed stochastic/saliency-based occlusion (Fong and Vedaldi, 2019; Zhong et al., 2020), or automatically modifying the background (Ryali et al., 2021). 3) Color-oriented transformations can be the selection of color channels (Tian et al., 2020) or Gaussian noise (Chen et al., 2020). A related line of work also proposes to learn how to generate/select data-augmentations (Cubuk et al., 2018; Tian et al., 2020). We expect that actively selecting what to perceive, typical of how an infant interacts with objects (James et al., 2001; Smith et al., 2018), will further improve the benefits of our natural augmentations.

**Time-based data-augmentations.** Several works proposed to use the temporality of interactions to build representations, but they consider smaller sets of augmentations or do not apply it to solve category recognition. A recent line of work proposes to learn embeddings of video frames using the temporal contiguity of frames: Knights et al. (2021) propose a learning objective that makes codes of adjacent frames within a video clip similar, however the system still needs to have information about where each video starts and ends. In contrast, both of our setups expose the system to a continuous stream of visual inputs much more like Orhan et al. (2020). But instead of trying to segment the stream into temporal classes our approach focuses on the representation of object categories. Moreover the two mentioned approaches do not analyze the impact of each interaction type, do not consider fundamental properties of vision such as a finite depth of field and datasets are restricted to few weeks of interactions.

A large body of work considers time-contrastive learning in the context of reinforcement learning (Oord et al., 2018; Laskin et al., 2020; Stooke et al., 2021; Okada and Taniguchi, 2021) and intrinsic motivation (Guo et al., 2021; Yarats et al., 2021; Li et al., 2021; Aubret et al., 2021), but they do not take into account interactions with objects. Other approaches look at natural data-augmentations. Schneider et al. (2021) showed the importance of object rotations, elevations, distance changes, for object recognition, but only considers small datasets of few objects, without complex backgrounds and without studying generalization over categories of objects. Wang et al. (2021) have shown that one can replace crop augmentations by saccade-like magnifications of images, but they use datasets of static images, making impossible to truly simulate a view change. Aubret et al. (2022) also showed that embodied visual inputs favor object recognition with time-contrastive learning. They also highlight that complex backgrounds negatively impact the representations of objects. In contrast, we show that allowing the agent to move with objects so that they are seen against changing backgrounds improves robustness against background complexity.

### 3 Methods

Our goal is to study the potential of natural interactions with objects for generating high-quality training data for time-contrastive learning and to evaluate the utility of the learned representations for object categorization. To dissect the roles of different time-based augmentations and to validate our approach on real-world video data, we instantiate two complementary training environments. The first environment, called Virtual Home Environment, enables us to precisely control the kind of interactions that feed our time-contrastive learning, and thus to properly disentangle their relative benefits. An agent can “play” with thousands of objects in a computer-rendered near-photorealistic house. The agent turns its body while continuing to look at the same object, switches the held object without turning its body, moves its gaze and adapts its focus distance (accommodation), turns, elevates and varies the distance of the objects.

The second environment allows us to validate our findings on real-world video footage. Here, we consider the CORe50 dataset of videos of first-person view object manipulations (Lomonaco and Maltoni, 2017). We refer to this environment as the CORe50 Environment. It incorporates realistic object manipulations and smooth visual tracking of objects. By dynamically reordering the frames of videos, we study how the resulting temporal structure of interactions shapes meaningful object representations in a time-contrastive learning framework. In the next sections, we describe both setups in detail.
3.1 Virtual Home Environments

Object interactions in near photo-realistic houses. To simulate the experiences of an infant in its home, we take inspiration from (Aubret et al., 2022) and place an embodied agent within an environment that resembles a residential house, where it can “play” with objects and change positions. Figure 1 depicts a bird’s eye view showing the floor plan of aforementioned house. In comparison with Aubret et al. (2022), we integrate a two order of magnitude higher number of toy objects as well as novel interactions with these objects. We also introduce novel houses, used to test the robustness of the object-representation with respect to novel backgrounds.

Toys4k dataset. The Toys4k dataset is composed of 4,179 diverse 3D toy objects distributed in an unbalanced way into 105 categories (Stojanov et al., 2021). It was designed to match the objects typically encountered by infants. We import two versions of the objects into TDW. First, to study the effect of time-based augmentations on shape generalization, we import untextured versions of the objects. Second, to consider the effects of object colors, we modify/remove 3D models, only keeping quick-to-process and good-quality objects in our simulations (cf. Appendix C.1 for more details). In both cases, to study the ability of our agent to generalize over categories, we use two thirds of the objects of each category for training and keep the last third for testing. To obtain a fair and sound test set, we generate 3 randomly rotated views of the test objects when the agent does not move and 5 when it moves in a house environment (see section on ego-motion below). This aims to account for the diversity of backgrounds. Except for rotation, no additional transformations have been applied. To assess the impact of our augmentations, we setup two toys environments in addition to the full house environment. In the first one, the agent interacts with grey objects in front of a fully white background. This marginalizes the possible impact of background and colors. Second, we let the agent interact with colored objects without moving. This aims to isolate the impact of interactions on the background.

Overview of the natural interactions The interactions take place on two timescales. First, the agent is spawned in a randomly sampled location and in a 100-timesteps-long episode (or session). Its position in the world determines what background will be seen. Each background contains a unique combination of static objects and floor/wall textures (cf. Appendix). Second, at every timestep the agent is presented with the choice of either turning its body towards another object (to the right or left) or keep interacting with the current object. These interactions are three-fold: through gaze control, object-oriented interactions and ego-motion. In humans, gaze control through saccades occurs on a faster timescale than object control/ego-motion; therefore, we assume the agent can instantaneously reset the gaze position while it only progressively changes the object in front of it and its own position.
Figure 2: Examples of time-based augmentations while “playing” in the virtual home environment (left, A-F): A) saccades; B) focus distance; C) elevation change; D) object rotation; E) distance change; F) turning with object so that it is seen against a different background. CORe50 (right, G-H): G) close-by frames of objects in 4 different recording sessions; H) different exemplars of the object category “cellphones” shown in different contexts.

Gaze control. First, to simulate saccadic eye movements (Trukenbrod et al., 2019), we enable the rotation of the camera towards a given angle around the object position. The apparent effect is a motion of both the background and the object (see Figure 2A). The agent uniformly samples pan and tilt angles in \([-A; A]\), where \(A\) is a hyper-parameter that defines the maximal saccade angle. Second, humans have a limited depth of field and need to adapt the shape of their lens to the distance of the fixation point, also called focus distance. As a consequence, the fixation point appears sharp in the visual field, while stimuli before and behind the fixation point are increasingly blurred. We model this by allowing the focus distance to range in \([0.47; 1.87]\) meters. We display the effect in Figure 2B. At object initialization or when this feature is disabled, we apply the perfect strategy of always focusing on the object.

Object-oriented manipulations. The agent can decide to replace the object being watched by another randomly chosen one. Thus, unlike Aubret et al. (2022), we do not pre-sample all objects of the location at the beginning of the episode. If it chooses to keep the same object instead, it can manipulate the object in three different ways. First, as shown in Figure 2C, it may lower or lift the object by an amount drawn uniformly from \([-0.1; 0.1]\) meters, with lower/upper bounds at 0 and 0.5 meters. Second, it can randomly rotate the object within an angle of \([0, S]\) degrees, with hyper-parameter \(S\) (cf. Figure 2D). We set the initial elevation to 0m (on the floor). Third, the agent can also bring the object closer or move it further away, as shown in Figure 2E; we model this as a change of distance uniformly sampled from \([-0.1, 0.1]\) meters, with the overall distance bounded between 0.6 and 1.2 meters. The bounds are intended to simulate the morphological limitations of the agent’s arms.

Ego-motion. To simulate an infant moving while holding an object, our agent can turn its body to a neighboring object position while keeping the same object. By avoiding displacement across possible agent locations, we keep the backgrounds temporally consistent, as typically experienced by humans. As shown in Figure 2F, it results in a change of background but often with similar floors and walls. We model this as turning every \(N_s\) steps in a deterministic way and switching object with probability \(p_o\). A small/medium \(N_s\) respectively simulates in an unnatural/natural fashion an agent that always/sometimes moves with its object. With the default behavior, the agent simultaneously picks up a new object and turns every 10 timesteps, otherwise we set \(p_o = 0.1\). For some experiments, we also allow the agent to visit another room every \(N_s\) steps while continuing to interact with the same object. It mostly highlights an extreme unnatural case for which the agent always makes large displacements rather than locally turning in the same room.

Evaluation procedure. To compare time-based augmentations to conventional image transformations we consider a widely used set of augmentations (Chen et al., 2020), i.e. crop/resize, Gaussian blur, grayscale, color jittering and horizontal flip with their default parameters. While Chen et al. (2020) randomly and independently applies each data-augmentation with a fixed probability, we do not implement particular acting strategies and always simultaneously apply our time-based augmentations.

To make the comparison fair in the simulated house, both are trained on the same amount of data: the agent collects images, augments them (next image for SimCLR-TT or a sequence of augmentation for SimCLR) and stores them in a buffer of size 100,000. When we study their combined effect, we
apply the sequence of augmentations of SimCLR to the next image in a sequence. It simultaneously trains on images randomly sampled from the buffer at every timestep. We make sure we compare the quality of the augmentations, and not the amount of data related to the test-set that feeds the CL algorithm. Thus, unless stated otherwise, we also rotate the object for the SimCLR baseline, since different rotations are present in the test set, and we use the same frequency of motion/room changes. We use a relatively high depth of field (distance range where objects produce a sharp image) to make sure SimCLR does not access less low-frequency information about the object than SimCLR-TT. We average all results over 3 random seeds. Hyper-parameters are given in the Appendix.

3.2 CORe50 Environment

The CORe50 dataset (Lomonaco and Maltoni, 2017) contains first-person views of an observer manipulating objects. The dataset comprises 167,866 images of $N_{obj} = 50$ different objects belonging to 10 distinct object categories. All objects were filmed in 11 sessions corresponding to different natural environments. For the training set we use images from all but one object per class and put aside every tenth image to form a separate validation set. For testing we use the images of the remaining object per class. This way we can efficiently test the generalization capabilities for object categories but also investigate how much of the background and object-specific information is encoded in the learnt representation.

Re-sampling of video frames Similar to a recent approach in (Schneider et al., 2021), we use a dynamic sampling procedure to re-order the video-frames of the training set into a plausible stream of object views. The resulting view sequence approximates object interactions that an infant might experience during early childhood. While generating these view sequences we distinguish two different ways of how the next view of the same object is sampled across a number of fixations $N_o$.

Our default method is called random walk view sampling. Here, the next view of an object is chosen to be one of the two adjacent video frames. The second method is the uniform view sampling. Here, the next view is picked uniformly at random from the same session. This approach provides the most diversity among successive views, which is expected to aid learning, but is not meant to simulate an infant’s viewing experience.

For every view the probability to stay on the same object is given by $p_o$. That means we model the number of fixations as a sequence of independent Bernoulli trials given a predefined probability, which yields

$$E[N_o|p_o] = 1 + \sum_{x=0}^{\infty} xp_o^x(1-p_o) = 1 + \frac{p_o}{1-p_o}$$

for the expected value of successive fixations on the same object. Both random walk and uniform view sampling are limited to their respective recording session. Therefore, in a similar manner, we also define $p_s$, which is the probability of sticking with the same session. Thus, for each new fixation there is a chance of $p_s$ that the resulting view will be taken from a different session. This cross session sampling (CSS) simulates the behaviour of taking an object to another room and naturally takes place on a slower timescale than the number of fixations. It is the analogue to room changes in the Virtual Home Environment.

The dynamic sampling approach enables us to investigate the influence of different temporal properties of the data in a controlled fashion by adapting the probabilities $p_o$ and $p_s$. The corresponding parameters $N_o$ and $N_s$ play an analogous role to choosing the frequency of object-oriented manipulations and ego-motion in the Virtual Home Environment.

Building the training buffer is done in cycles, see Figure 3A. In every cycle each of the $N_{obj}$ objects is chosen exactly once — the order of the objects is a new random permutation during each cycle. Thus, a cycle consists of a total of $N_{obj} \times N_o$ object views. During the fixation on one object multiple cross-session transitions can occur, see Figure 3B. Several cycles of $N_{obj} \times N_o$ views are stored after another. During training, batches of pairs of subsequent views are sampled uniformly from the buffer (dashed brackets) and filled into the batch. We consider an epoch to be an entire run through the buffer, which is chosen to match the size of the underlying CORe50 data set. In consequence, every training image on average is presented once during each epoch.
4 Results

In this section, we compare time-contrastive learning from natural interactions to conventional data augmentations and provide an in-depth analysis of the impact of object rotations and ego-motion on the learned representations.

4.1 Time-based augmentations derived from natural interactions boost performance of contrastive learning

In Figure 4A, we compare several variants of SimCLR-TT against SimCLR and a combination of both in the Virtual Home Environment. Baseline for comparison is SimCLR with its standard augmentations (black curve). Adding frequent room changes (purple curve) does not affect performance. The green (rot(30)+D+Ns=2) variant simulates an infant that plays in one room, turns regularly, slowly rotates objects, brings them closer/farther and use SimCLR augmentations. The blue curve (rot(360)+D+Ro+Ns=1) displays an extreme case of a simulated infant that always changes rooms while very quickly rotating objects. In red (rot(360)+D+Ro+Ns=1+SimCLR), we observe the same extreme case, but combined with SimCLR augmentations. We clearly see that the best variant of SimCLR-TT outperform SimCLR by a large margin and that both can be combined to further increase the overall performance.

For the CORe50 Environment we observe that the SimCLR-TT approach can outperform conventional SimCLR and supervised learning on the validation set. For random walk view sampling Figure 4B shows a steady increase of performance with the number of fixations, which is expected. When comparing both view sampling approaches the uniform view sampling especially helps in generalizing to unseen video frames featuring the same sessions and objects as the training data, i.e., the validation set. However, the SimCLR-TT approach struggles to generalize seen objects to an unseen object of the same category on its own, as seen in Figure 4C. The generalization capabilities do not exceed the conventional SimCLR approach. Note that the uniform sampling (orange) does perform better than the random walk (blue), which is consistent with our experiments regarding object rotation in the Virtual Home Environment.

Combining the standard SimCLR augmentations with time-contrastive augmentations substantially improves the generalization capabilities (dashed curves), by revealing information about the 3D structure of objects and placing the same object within different backgrounds. We also did an ablation on different values for the frequency of session changes and indicate peak performance with a star, cf. appendix B.2.

In the next sections, we will analyze the impact of rotations and motion speed on the learnt representations. For other augmentation methods, in the Appendix, we emphasize the contribution of each time-based and conventional augmentation for category recognition. We also discuss how saccades relate to the conventional crop/resize data-augmentation in the Appendix.

4.2 Object manipulations including 3-D rotations support shape generalization

To assess the importance of object rotations to generalize over object shapes, we conducted experiments in the Virtual Home Environment. We let the agent interact with untextured (grey) versions of the objects, which are seen against a white background. Figure 5A shows that increasing the speed of object rotations systematically increases the category recognition accuracy. Interestingly, we observe the inverse effect on individual object recognition accuracy (Figure 5B). We conclude that fast object rotations (in the extreme case constructing positive pairs from randomly rotated views of an object)
Figure 4: A. Comparison between SimCLR-TT variants, SimCLR and the combination of both for the Virtual Home Environment. Time-based augmentations are labeled by the changes they induce: D=depth, Ro=room changes. Shaded areas represent one standard deviation (over different random seeds). B and C: Category accuracy for the CORE50 Environment, $p_s = 0.95$. B. Validation set accuracy compared to the baselines supervised (grey) and SimCLR (purple). C. Same for test set. Shaded areas represent one standard deviation of test category accuracy over seeds.

Figure 5: A. Test category recognition accuracy according to the maximal rotation speed $S$, without other augmentations. B. Test object recognition accuracy according to the maximal rotation speed $S$, without other augmentations. C. Test category accuracy according to different combinations of flip and rotation speeds. We used grey objects in front of a blank background. The shaded area displays the +/- standard deviation of test category accuracy over seeds.

tend to discard object-specific details and focus the representation on the overall shape of the object, which will be similar for different objects of the same category.

Next, we wanted to understand how object rotations relate to the standard horizontal flip augmentation, since the flip resembles a 180-degrees rotation for symmetric objects and uniform background. In Figure 5C, we combined the flip augmentation and object rotations with other SimCLR augmentations. We see that when the object rotation speed is large enough ($S = 360$), there is no additional benefit in using the horizontal flip augmentation. However, rotations always boost performance compared to the pure flipping augmentation. Thus, we conclude that using 3-D object rotations rather than flipping results in considerably better representation of object shape.

4.3 Moving with objects prevents the backgrounds from cluttering the representation.

To evaluate the impact of ego-motion while holding objects, we trained our agent in the Virtual Home Environment with different levels of motion. In Figure 6A), we first see that increasing the frequency $N_s$ and the range (room) of motion considerably boosts the test category accuracy in backgrounds from the trained house. At the same time, we observe a decrease of the accuracy when classifying the background Figure 6B). We conclude that moving while holding objects improves the representation’s invariance with respect to background. However, we also observe that these two effects are less pronounced for novel backgrounds (blue) from three unfamiliar houses. In contrast, we see in Figure 4A that the quantity of motion does not significantly impact the SimCLR performance (compare black curve with purple curve).
We investigated the impact of time-based, infant-inspired natural object interactions as a data-augmentation method on a representation learnt through CL. We studied these interactions through a novel 3-D simulation environment where an agent “plays” with objects in a house and we validated our findings on the CORe50 dataset of real-world videos of object manipulations. We find that combining time-based augmentations with conventional data-augmentations greatly improves the generalization and, in line with previous work (Wang et al., 2021), we highlighted that saccades can have a similar effect as random cropping. However, we found that color-based transformations like color jittering and grayscaling remain important. While we expect that direct and indirect changes of lighting could simulate this in a time-contrastive approach, the TDW platform does not support ray-tracing, which would allow modeling such illumination changes realistically.

Our work raises the question whether time-based augmentations could fully replace conventional ones. Indeed, we showed that 3-D rotations are superior to the flip data-augmentation for shape generalization and, in line with previous work (Wang et al., 2021), we highlighted that saccades can have a similar effect as random cropping. However, we found that color-based transformations like color jittering and grayscaling remain important. While we expect that direct and indirect changes of lighting could simulate this in a time-contrastive approach, the TDW platform does not support ray-tracing, which would allow modeling such illumination changes realistically.

Our work has a number of limitations. First, while we have portrayed the temporal augmentations as stemming from object manipulations, we did not render a hand holding the object or similar. Second, the CORe50 dataset is small. Experiments on larger dataset with richer and more controlled forms of object manipulations may provide additional insights. Finally, in this work, we only considered naive behavioral strategies for, e.g., lifting and turning objects or ego-motion. This contrasts with the way...
infants learn about their environment (Bambach et al., 2016). For example, they are biased towards creating planar views of objects (Pereira et al., 2010), synchronize their saccades and focus distance according to the fixation point and often prefer unfamiliar objects (Roder et al., 2000). Thus, we expect that learning to actively select successive views will further unveil the potential of time-based augmentations.

Acknowledgements

This work was sponsored by a public grant overseen by the French National Agency through the IMobS3 Laboratory of Excellence (ANR-10-LABX-0016) and the IDEX-ISITE initiative CAP 20-25 (ANR-16-IDEX-0001). Financial support was also received from Clermont Auvergne Metropole through a French Tech-Clermont Auvergne professorship. We gratefully acknowledge support from the CNRS/IN2P3 Computing Center (Lyon - France) for providing computing and data-processing resources needed for this work. This work was also performed using HPC resources from GENCI–IDRIS (Grant 2022-AD011011623R1). JT was supported by the Johanna Quandt foundation.

References

Aubret, A., Hassas, S., et al. (2021). Distop: Discovering a topological representation to learn diverse and rewarding skills. arXiv preprint arXiv:2106.03853.

Aubret, A., Teulière, C., and Triesch, J. (2022). Embodied vision for learning object representations. arXiv preprint arXiv:2205.06198.

Bambach, S., Crandall, D. J., Smith, L. B., and Yu, C. (2016). Active viewing in toddlers facilitates visual object learning: An egocentric vision approach. In CogSci.

Bardes, A., Ponce, J., and Lecun, Y. (2022). Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In ICLR 2022-10th International Conference on Learning Representations.

Byrge, L., Sporns, O., and Smith, L. B. (2014). Developmental process emerges from extended brain–body–behavior networks. Trends in cognitive sciences, 18(8):395–403.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In International conference on machine learning, pages 1597–1607. PMLR.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. (2018). Autoaugment: Learning augmentation policies from data. arXiv preprint arXiv:1805.09501.

Fong, R. and Vedaldi, A. (2019). Occlusions for effective data augmentation in image classification. In 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), pages 4158–4166. IEEE.

Gan, C., Schwartz, J., Alter, S., Mrowca, D., Schrimpf, M., Traer, J., De Freitas, J., Kubilius, J., Bhandwaldar, A., Haber, N., et al. (2021). Threedworld: A platform for interactive multi-modal physical simulation. In Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1).

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., et al. (2020). Bootstrap your own latent: A new approach to self-supervised learning. arXiv preprint arXiv:2006.07733.

Guo, Z. D., Azar, M. G., Saade, A., Thakoor, S., Piot, B., Pires, B. A., Valko, M., Mesnard, T., Lattimore, T., and Munos, R. (2021). Geometric entropic exploration. arXiv preprint arXiv:2101.02055.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
Inoue, H. (2018). Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929*.

Jaiswal, A., Babu, A. R., Zadeh, M. Z., Banerjee, D., and Makedon, F. (2021). A survey on contrastive self-supervised learning. *Technologies*, 9(1):2.

James, K. H., Humphrey, G. K., and Goodale, M. A. (2001). Manipulating and recognizing virtual objects: where the action is. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*, 55(2):111.

Knights, J., Harwood, B., Ward, D., Vanderkop, A., Mackenzie-Ross, O., and Moghadam, P. (2021). Temporally coherent embeddings for self-supervised video representation learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 8914–8921. IEEE.

Laskin, M., Srinivas, A., and Abbeel, P. (2020). Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pages 5639–5650. PMLR.

Li, N. and DiCarlo, J. J. (2010). Unsupervised natural visual experience rapidly reshapes size-invariant object representation in inferior temporal cortex. *Neuron*, 67(6):1062–1075.

Li, S., Zheng, L., Wang, J., and Zhang, C. (2021). Learning subgoal representations with slow dynamics. In *International Conference on Learning Representations*.

Lomonaco, V. and Maltoni, D. (2017). Core50: a new dataset and benchmark for continuous object recognition. In *Conference on Robot Learning*, pages 17–26. PMLR.

Loshchilov, I. and Hutter, F. (2018). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Misra, I. and Maaten, L. v. d. (2020). Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717.

Noroozi, M. and Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer.

Okada, M. and Taniguchi, T. (2021). Dreaming: Model-based reinforcement learning by latent imagination without reconstruction. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4209–4215. IEEE.

Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Orhan, E., Gupta, V., and Lake, B. M. (2020). Self-supervised learning through the eyes of a child. *Advances in Neural Information Processing Systems*, 33:9960–9971.

Pereira, A. F., James, K. H., Jones, S. S., and Smith, L. B. (2010). Early biases and developmental changes in self-generated object views. *Journal of vision*, 10(11):22–22.

Roder, B. J., Bushnell, E. W., and Sasseville, A. M. (2000). Infants’ preferences for familiarity and novelty during the course of visual processing. *Infancy*, 1(4):491–507.

Ryali, C., Schwab, D. J., and Morcos, A. S. (2021). Learning background invariance improves generalization and robustness in self-supervised learning on imagenet and beyond.

Schneider, F., Xu, X., Ernst, M. R., Yu, Z., and Triesch, J. (2021). Contrastive learning through time. In *SVRHM 2021 Workshop @ NeurIPS*.

Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48.

Smith, L. B., Jayaraman, S., Clerkin, E., and Yu, C. (2018). The developing infant creates a curriculum for statistical learning. *Trends in cognitive sciences*, 22(4):325–336.
Stojanov, S., Thai, A., and Rehg, J. M. (2021). Using shape to categorize: Low-shot learning with an explicit shape bias. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1798–1808.

Stooke, A., Lee, K., Abbeel, P., and Laskin, M. (2021). Decoupling representation learning from reinforcement learning. In International Conference on Machine Learning, pages 9870–9879. PMLR.

Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., and Isola, P. (2020). What makes for good views for contrastive learning? Advances in Neural Information Processing Systems, 33:6827–6839.

Trukenbrod, H. A., Barthelmé, S., Wichmann, F. A., and Engbert, R. (2019). Spatial statistics for gaze patterns in scene viewing: Effects of repeated viewing. Journal of Vision, 19(6):5–5.

Tsai, Y.-H. H., Wu, Y., Salakhutdinov, R., and Morency, L.-P. (2020). Self-supervised learning from a multi-view perspective. In International Conference on Learning Representations.

Wang, B., Mayo, D., Deza, A., Barbu, A., and Conwell, C. (2021). On the use of cortical magnification and saccades as biological proxies for data augmentation. In SVRHM 2021 Workshop@ NeurIPS.

Wiskott, L. and Sejnowski, T. J. (2002). Slow feature analysis: Unsupervised learning of invariances. Neural computation, 14(4):715–770.

Wood, J. N. and Wood, S. M. (2018). The development of invariant object recognition requires visual experience with temporally smooth objects. Cognitive Science, 42(4):1391–1406.

Yarats, D., Fergus, R., Lazaric, A., and Pinto, L. (2021). Reinforcement learning with prototypical representations. arXiv preprint arXiv:2102.11271.

Zhong, Z., Zheng, L., Kang, G., Li, S., and Yang, Y. (2020). Random erasing data augmentation. In Proceedings of the AAAI conference on artificial intelligence, volume 34, pages 13001–13008.
Ablation of data-augmentations in Virtual Home Environment

Positive/negative impacts of different data-augmentations. In Figure 8, we test different sets of augmentations in the Virtual Home Environment, progressively combined in a cumulative way. Interestingly, not all augmentations lead to performance improvements. There are two notable exceptions. First, the elevation of the object (E) consistently has a negative impact on the representation. We hypothesize that this is because a too large elevation removes important parts of the objects from the field of view. We cannot rule out the possibility that smaller elevation changes could improve performance. Second, the focus distance changes (F) also hurt the performance of the representation in the Virtual Home Environment. As our parameters tend to more easily blur the object rather than the background, we hypothesize it may bias the representation towards background-oriented features.

We also notice that color jittering (J) is vital for category recognition, in particular when the objects are colored (blue and green curves). We deduce it plays a crucial role to learn the color invariance of a category. The other augmentations at least do not significantly hurt the representation, and at best improve it. We leave the analysis of rotations to §4.2 and we analyse the effect of saccades below.

Saccades/depth changes have a similar effect as crop/resize. We want to assess the impact of saccades on shape generalization. We conduct a series of experiments in the Virtual Home Environment with gray objects in front of white backgrounds. We first look for the best maximum range of saccades in Figure 9A; we clearly observe a sweet spot at 24 degrees, with an accelerated convergence when saccades are combined with depth changes. We qualitatively observe in Figure 10 that a maximum angle of 24 degrees corresponds to objects being cropped in the middle. To check if this saccade-based crop can replace the crop/resize augmentation, we test different combinations of saccade and crop/resize in Figure 9. We observe that their respective effects do not simply add but rather suggest a similar effect, with crop/resize outperforming all saccade-based experiments. To understand if this is due to the fact that the agent always uniformly samples saccades in $[-A; A]$, we test another sampling strategy $sac(o)$. In this case, the agent always samples a 24-degrees angle (pan and tilt) with a probability of 0.5, and keeps the object centered otherwise. With $sac(o)$, we see in Figure 9C that the agent manages to asymptotically achieve similar performance as for the crop/resize augmentation. However, we observe in Figure 10A that this replacement is not possible when considering diverse backgrounds. We hypothesize that saccades are more sensitive to backgrounds.
Figure 9: Test category accuracy according to: A. different maximum saccade amplitudes, without other augmentations; B. different combinations of saccades with maximum angle of 24 degrees, and crop/resize augmentations, where all curves implement all SimCLR augmentations (except crop/resize). C. Same as B. but using the \textit{sac(o)} strategy. We used grey objects in front of a white background for these experiments.

Figure 10: A. different combinations of saccades and crop/resize augmentations, where all curves implement all SimCLR augmentations (except crop/resize) in the Virtual Home Environment. B. Examples of maximal amplitude $A$ of pan-oriented saccades.

than crops are. This might be because the eyes’ motion causes new portions of the background to become visible. The greater amount of seen backgrounds may clutter the representation.

\textbf{Color transformations and high frequency of room changes have complementary effects.} Here, we study the interplay between color invariance (color transformations) and background invariance (high-frequency turns with room changes) in the Virtual Home Environment. We see in Figure\textsuperscript{[1]} that adding color jittering, grayscaling, Gaussian blur and horizontal flip data-augmentations to high-frequency room changes considerably reduces the standard deviation of accuracy over novel houses. In addition, unlike in Figure\textsuperscript{[6]}, adding frequent room changes significantly increases the average accuracy on novel houses when color transformations are also used. We conclude that color transformations are crucial to make the representation invariant to a large set of backgrounds.

\section{B Complementary analysis in CORe50 Environment}

\subsection{B.1 Detailed analysis varying the number of fixations and session frequency}

To complement the analysis of results from the CORe50 Environment for the single training/testing split presented in the main manuscript (see Figure\textsuperscript{[4]}, we also evaluate the different algorithms on four additional splits of the training and validation sets, basically performing a five-fold cross validation. The mean and standard errors of that analysis can be found in Table\textsuperscript{[12]} We observe qualitatively similar results to those from the main text: the uniform view sampling (uni) performs best on the validation set whereas the two combined methods deliver the best generalization capabilities across splits. Figure\textsuperscript{[15]} shows the results in the form of box-plots.
Figure 11: Final Test category accuracy on all test houses and the training house according to different combinations of frequency of motion/room changes and color jittering, grayscaling, blur and horizontal flip. We also use other SimCLR-TT augmentations as in Figure 6. The accuracy was reported after 280,000 timesteps. We note: J=Color Jittering; G=Grayscale; B=Blur; H=Horizontal flip.

Figure 12: CORe50 Environment. Box plot of test set accuracy distribution over 5 different train/test splits after 100 epochs of training, $p_o = 0.95$, $p_s = 0.95$. Whiskers indicate min and max values.

We also report means and standard errors for category (Table 2) and session accuracy (Table 3) of our randomwalk algorithm when evaluated with different values for $p_s$ and $p_o$. Compared to the evaluation on a single split in the main text we observe slightly lower accuracies overall, but the same qualitative patterns. The standard error reveals that evaluation on the validation set is fairly consistent over different training splits, while test category accuracy varies.

B.2 Influence of session changes on combined augmentations.

Since our main analysis with the CORe50 Environment aims to give a solid overview over the presented approaches we have picked a medium level of session changes as the default ($p_s = 0.95$, $\mathbb{E} [N_s | p_s] = 20$). The results show that the combined approaches outperform the conventional SimCLR method, but we were also interested in how sensitive the novel methods are to the $p_s$ parameter. Results of that analysis are depicted in Figure 13. Especially the combined randomwalk (comb rw) method is sensitive to a lower number of fixations within the same session, which considerably improves validation set category accuracy (A). We assume that this is because the additional contrastive information is considerably higher for the randomwalk method, whereas uniform view sampling already achieves some degree of background variation due to sampling.
Table 1: Category accuracy and standard error on CORe50, \( p_s = 0.95 \). Standard error based on five different training/testing splits. Training occurred for 100 epochs, batchsize 512. Best performance is highlighted in bold, cf. Figure 4B, C.

|           | \( E [N_o] \) |
|-----------|---------------|
|           | 2  | 5  | 10 | 20 | 50 |
| supervised| .980 ± .017 |    |    |    |    |
| SimCLR    | .804 ± .111 |    |    |    |    |
| rw        | .430 ± .006 | .630 ± .005 | .810 ± .007 | .907 ± .005 | .939 ± .003 |
| uni       | .996 ± .001 | 1.000 ± .000 | 1.000 ± .000 | 1.000 ± .000 | 1.000 ± .000 |
| comb rw   | .757 ± .005 | .891 ± .004 | .940 ± .003 | .961 ± .002 | .971 ± .001 |
| comb uni  | .997 ± .001 \( \dagger \) | .998 ± .000 | .998 ± .000 | .999 ± .000 | .999 ± .000 |

Table 2: Category accuracy and standard error on CORe50, randomwalk sampling procedure. Standard error based on three different training/testing splits. Training occurred for 100 epochs, batchsize 512, cf. Figure 7A, B.

|           | \( E [N_o] \) |
|-----------|---------------|
|           | 2  | 5  | 10 | 20 | 50 |
| supervised| .530 ± .080 |    |    |    |    |
| SimCLR    | .544 ± .030 |    |    |    |    |
| rw        | .229 ± .018 | .327 ± .033 | .386 ± .046 | .429 ± .055 | .427 ± .052 |
| uni       | .426 ± .055 | .435 ± .060 | .440 ± .056 | .450 ± .059 | .440 ± .056 |
| comb rw   | .501 ± .032 | .589 ± .030 | .615 ± .029 | .629 ± .031 | .638 ± .032 |
| comb uni  | .605 ± .045 \( \dagger \) | .608 ± .036 | .620 ± .036 | .616 ± .036 | .616 ± .033 |

\( \dagger \) based on 3 training/testing splits

contrasts from one whole session compared to the neighboring frames. For testset accuracies we observe a similar trend albeit not as pronounced (B). Note how test-accuracies for the random walk procedure are considerably higher than for the uniform sampling. In line with results from our Virtual Home Environment experiments we observe that for very low values of \( N_o \) the internal representation contains less information about the session for the randomwalk approach (C,D). The uniform method, however, shows robustness to this effect.

16
Table 3: Session accuracy and standard error on CORe50, SimCLR-TT algorithm with randomwalk sampling procedure. Standard error based on three different training/testing splits. Training occurred for 100 epochs, batchsize 512, cf. Figure 7C, D.

| Validation set | E $[N_o]$ |
|----------------|-----------|
|                | 2 5 10 20 50 |
| 2              | .596 ± .004 | .565 ± .008 | .540 ± .011 | .523 ± .005 | .545 ± .012 |
| 5              | .717 ± .001 | .662 ± .009 | .671 ± .004 | .690 ± .004 | .705 ± .005 |
| 10             | .688 ± .011 | .717 ± .011 | .748 ± .005 | .741 ± .003 | .736 ± .002 |
| 20             | .711 ± .015 | .708 ± .005 | .758 ± .006 | .769 ± .003 | .761 ± .002 |
| 50             | .732 ± .002 | .737 ± .005 | .756 ± .009 | .762 ± .006 | .769 ± .010 |

| Test set       | E $[N_o]$ |
|----------------|-----------|
|                | 2 5 10 20 50 |
| 2              | .572 ± .010 | .535 ± .011 | .540 ± .015 | .507 ± .008 | .518 ± .010 |
| 5              | .665 ± .012 | .644 ± .014 | .658 ± .018 | .671 ± .011 | .704 ± .011 |
| 10             | .646 ± .016 | .695 ± .019 | .718 ± .016 | .718 ± .014 | .707 ± .009 |
| 20             | .670 ± .026 | .673 ± .003 | .716 ± .021 | .740 ± .013 | .735 ± .013 |
| 50             | .704 ± .012 | .689 ± .025 | .732 ± .012 | .728 ± .015 | .731 ± .020 |

Figure 13: Analysis of the combined augmentation approach for different values of $p_s$, CORe50 Environment. Error bars indicate standard error based on three different training/testing splits.

C  Assets and evaluation in the Virtual Home Environment

C.1 3D objects

We import two versions of objects into the ThreeDWorld Software (TDW):

Untextured objects: this is the grey version of objects. We export the 3D models to .obj format and remove the texture files. We discard objects that took too long to process in TDW.

Textured objects: this is the colored version of objects. In order to import them in TDW, we apply a series of operators: 1) we reduce the complexity of some meshes to make the next step computationally feasible and reduce the TDW processing complexity; 2) we manually bake most of the 3D models to obtain a single texture file with the meshes; 3) we import the models into TDW; 4) we resize the models to marginalize the effect of size on category recognition and avoid harmful collisions. We visually check the quality of all models. When we do not manage to obtain good-looking textures for some objects, we remove them from our dataset (< 10% of the dataset). The tree category does not contain enough good-looking objects, so we remove the whole category. We provide the TDW libraries of objects at https://zenodo.org/record/6580588#.Yo8qAjnF1H4
Figure 14: Test images of an airplane. A. without texture in front of white background; B. with texture in front of one background; C. with texture in the training house.

C.2 Images from the test datasets.

Our House test set is composed of 7,740 images, from 1,548 objects distributed into 105 categories. In Figure 14, we show examples of images of an airplane used in the test datasets. We used a different depth of field value than the default training one, making the background more blurry than during training. We expect it makes the classification on the test set slightly harder for all methods. Note that test-like images were not accessible through changes of focus distance. By showcasing the category “elephant” in Figure 15 we also exhibit the diversity of objects within one category. The elephant category includes flying elephants, bipedal elephants, red elephants, differently positioned elephants or geometrically simplified elephants.

In Figure 16, we display the three test houses used in Figure 6 and Figure 8. House scenes were taken from pre-provided bundles in TDW and we added the agent and objects in the scene similarly to Figure 1.

We provide the test sets in https://zenodo.org/record/6580588#.Yo8qAjnP1H4. The underlying ThreeDWorld (TDW) software used is licenced under BSD 2-Clause.

C.3 Evaluation of the representation.

To assess the quality of our representation, we train a linear readout on top of the representation in a supervised fashion, as previously done (Chen et al., 2020). Because of the limited buffer size, the training data changes over time in the Virtual Home Environment. To limit the impact of such shift on the evaluation, we train the linear evaluation classifier on the test set. While the learned linear classifier could be different from a classifier learned on the training data, it still assesses the linear separability of categories. It also makes possible to evaluate the linear separability of unseen objects in Figure 5B.

D Hyper-parameters

For all conducted experiments we apply a weight decay of $10^{-6}$ and update weights with the AdamW optimizer (Loshchilov and Hutter, 2018) and a learning rate of $5 \cdot 10^{-4}$.

Virtual Home Environment. The agent in the Virtual Home Environment perceives the world around it as $128 \times 128$ RGB images. These images are encoded by a succession of convolutional layers with the following [channels, kernel size, stride, padding] structure: [64, 8, 4, 2], [128, 4, 2, 1], [256, 4, 2, 1], [256, 4, 2, 1]. The output is grouped by an average pooling layer and a linear layer ending with 128 units. Each convolution layer is followed by a non-linear ReLU activation function and a dropout layer ($p = 0.5$) to prevent over-fitting. Our preliminary experiments exhibited that such a shallow architecture combined with cosine similarity works better in this setting than standard architectures used in contrastive learning (Chen et al., 2020). We use a batch size of 512 and a buffer size of 100,000. Average training time was 48 hours per run. All experiments ran on GPUs of type NVIDIA V100.
Figure 15: All objects of the category “elephant.” used in our Virtual Home Environment.

Figure 16: Top view of houses, without ceiling, used only when testing category recognition in unfamiliar houses experiments. A. House 2; B. House 3; C. House 4.
**CORe50 Environment.** The additional diversity of the CORe50 training data required a more capable encoder network, which we chose to be a ResNet-18 [He et al., 2016]. The encoder transforms the input images into a 128-dimensional latent representation, which is followed by a 2-layer linear projection head [256, 128]. For the projection head we use batch normalization and a ReLU activation function in-between the two layers. We chose a batch size of 512 and training was done for 100 epochs on a GPU of type NVIDIA V100 or NVIDIA RTX2070 SUPER. Average training time was 17.5h per run. The CORe50 dataset is licensed under CC-BY-4.0.