TIME TO AUGMENT VISUAL SELF-SUPERVISED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Biological vision systems are unparalleled in their ability to learn visual representations without supervision. In machine learning, self-supervised learning (SSL) 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 SSL, 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 manipulations 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 self-supervised learning (SSL) acquire representations invariant to data-augmentations based on simple image manipulations like crop/resize, blur or color distortion (Grill et al., 2020; Chen et al., 2020). 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, appear to 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 SSL approaches has led to recent 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). How these systems generalize depends on the temporal structure of their visual input. In particular, visual input arising from embodied interactions with objects may lead to quite different generalizations compared to what is possible with simple image manipulations. For instance, human infants learning about objects interact with them in various ways (Smith et al., 2018). First, they execute saccadic eye movements 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, 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. We refer to (simulations of) such interactions as *natural interactions*.

Here, we systematically study the impact of such natural interactions on representations learnt through time-contrastive learning in three settings. First, we introduce two new simulation environments based on the near-photorealistic simulation platform ThreeDWorld (TDW) (Gan et al., 2021) and combine them with a recent dataset of thousands of 3D object models (Toys4k) (Stojanov et al., 2021). Second, we validate our findings on two datasets of realistic images. The ToyBox dataset

validates the importance of object manipulations Wang et al. (2018) and we systematically study with the CORe50 dataset (Lomonaco and Maltoni, 2017) the impact of relative frequencies of object manipulations and ego-motion on the learned representation.

Our experiments show that adding time-based augmentations to conventional data-augmentations considerably improves category recognition. Furthermore, we show 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. We conclude that exploiting natural interactions via time-contrastive learning greatly improves self-supervised visual representation learning.

2 Related work

Data-augmented self-supervised learning. Recent developments in self-supervised learning have shown to successfully learn visual representations suitable to downstream tasks (Chen et al., 2020). The general idea of most of the recent approaches 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. The approach works as follows: 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. The initial and positive images are embedded through a Siamese-like architecture. Embeddings of positive pairs are brought together while keeping the embeddings of the batch overall different. There are three main categories of approaches which differ according to how inputs are made distant: contrastive learning methods Chen et al. (2020); He et al. (2020) explicitly push away the embeddings of a batch from each other; distillation-based methods Grill et al. (2020); Chen and He (2021) use an asymmetric embedding architecture, allowing to discard the "push away" part; entropy maximization methods Bardes et al. (2022); Ermolov and Sebe (2020) impose a high entropy over embedding values.

Visual data-augmentations. Most self-supervised learning approaches have used augmentations based on simple image manipulations to learn representations. Frequently used 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). 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), but since it takes advantage of labels, the approach is no longer self-supervised.

Time-based data-augmentations. Several works proposed to use the temporality of interactions to build representations, but they consider smaller sets of interactions 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, 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 different interaction types and do not consider fundamental aspects of vision such as a finite depth of field.

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 considered 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-based SSL 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 setups.

Two environments enable us to precisely control the kind of objects and the kind of interactions that feed our time-based SSL algorithm, and thus to properly disentangle their relative benefits. In the first environment, called *Virtual Home Environment* (VHE), an agent can "play" with thousands of texured 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 and varies the distance of the objects. The second environment called *3D Shape Environment* (3DShapeE), replicates the first one, but incorporates untextured objects and a blank background. It aims to study the impact of natural interactions on shape generalization only.

Two additional datasets allow us to validate our findings on real-world object footage. First, we test our method on the ToyBox dataset Wang et al. (2018) to validate our findings on first-person view multifaceted manipulations of objects. Second, we consider the CORe50 dataset of videos of first-person view object manipulations (Lomonaco and Maltoni, 2017) and 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-based self-supervised learning framework. In the next sections, we describe our setups in detail[1].

3.1 VIRTUAL HOME ENVIRONMENTS

Object interactions in near photo-realistic houses. To simulate the natural interactions 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 fully textured toy objects as well as novel interactions with these objects.

The agent is spawned in a randomly sampled location, staying there for 100-timestep-long episodes (or sessions). Its position in the world determines what background will be seen. Each background contains a unique combination of static objects and floor/wall textures.

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 place textured objects in different locations of the full house. We modify/remove 3D models, only keeping quick-to-process and correct-quality objects in our simulations (cf. Appendix C.1 for more details). 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 5 randomly rotated views of the test objects in different locations of the house. We verify in Appendix B.2 that this does not bias our evaluation.

[1]The corresponding source code will be made publicly available.

Figure 1: Top view of the VHE used to situate our agent in a house. Blue triangles/orange circles show possible agent/object positions, respectively. In this episode, the agent is located in the office (blue arrow) and interacts with a plush toy. Successive first person views of object interactions are shown in the lower left.

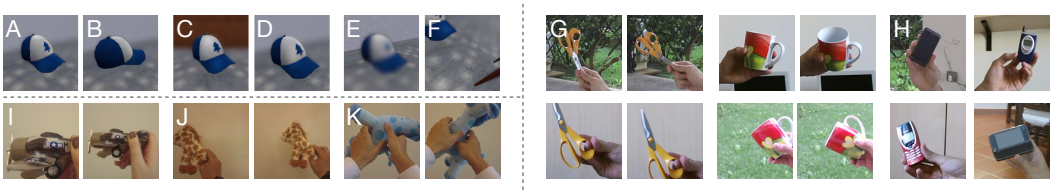

Figure 2: Examples of time-based augmentations while "playing" in the VHE (left, A-F): A) Initial image; B) object rotation; C) ego-motion; D) depth change; E) change of focus distance; E) saccadic eye movement. CORe50 (right, G-H): G) close-by frames of objects in 2 different recording sessions; H) different exemplars of the object category "cellphones" shown in different contexts. ToyBox (left, I-K): I) depth change, J) lateral translation, K) object rotation.

Except for rotation, no additional transformations have been applied. We also introduce novel houses to test the robustness of the object-representation with respect to novel out-of-distribution backgrounds (cf. Appendix C.2).

Overview of the natural interactions We want to simulate infant-like interactions, these include: 2) slow/quick object manipulations; 1) ego-motion with slow/quick background changes while holding the same object. These might contribute differently to the representation learnt through time-based SSL. We describe each of them below. We also discuss gaze control in Appendix A.1.

Object-oriented manipulations. Only a few months old, the infants start to hold objects and rotate them Byrge et al. (2014). To simulate this, when the agent keeps the same object (see below), it can manipulate the object in two different ways. First, it can randomly rotate the object within an angle of $[0, \text{rot}]$ degrees, with hyper-parameter rot (cf. Figure 2A,B). Second, the agent can also bring the object closer or move it further away, as shown in Figure 2A,D; we model this as a change of distance uniformly sampled from $[-0.075, 0.075]$ meters, with the overall distance bounded between 0.65 and 1.1 meters. The bounds intend to simulate the morphological limitations of the agent's arms.

Ego-motion. When infants learn to stand up and walk, they can play with objects while moving in their environment. To simulate this, 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 2A,C, 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 $1 - p_{\text{o}}$. A small/medium

N_s respectively simulates in a unnatural/natural fashion an agent that always/sometimes moves with its object. When this interaction is disabled, the agent simultaneously picks up a new object and turns every 10 timesteps, otherwise we set $p_\mathrm{o} = 0.9$. For some experiments (Table 1 and Figure 4B,C), 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 unnatural case for which the agent always makes large displacements rather than locally turning in the same room.

3.2 3D SHAPE ENVIRONMENT

In order to study the influence of natural interactions on shape generalization while marginalizing the impact of object colors or background, we introduce a simple 3D Shape Environment (3DShapeE). There are two differences with respect to the VHE: 1) we import untextured versions of toys objects from the Toys4k dataset, making them appear grey as shown in Appendix C.2; 2) we place the agent in an empty environment so that objects are seen against a blank background.

3.3 TOYBOX DATASET

The ToyBox dataset Wang et al. (2018) contains first-person video of an observer manipulating 360 objects (12 categories, 30 objects each). In each clip, a person applies one type of transformation: a translation (x, y or z), rotation (x, y or z), a hodgepodge, nothing specific or in some cases, nothing if there is no object. We sample two frames per second from each clip (cf.Appendix C.3 for more details). Unlike in the CORe-50 environment (see below), we only focus on the impact of object manipulations on time-based augmentations. Thus, we randomly sample images and time-based positive pairs are the next images in their video clip. At the end of a clip, the positive sample is the first image of the randomly sampled next clip.

3.4 CORE50 ENVIRONMENT

Overview. The CORe50 dataset (Lomonaco and Maltoni, 2017) contains first-person views of an observer manipulating objects. The dataset comprises 167,866 images of 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.

Using this dataset, we approximate different kind of natural interactions: 1) the number of fixations on an object; 2) ego-motion with slow/quick background changes while holding the same object; 3) The speed of object manipulation. We describe each of them below.

Number of fixations. Here we simulate the different amount of time an infant plays with an object. For every sampled image, the probability that the positive pair is from the same object is given by p_o. That means we model the number of fixations N_o as a sequence of independent Bernoulli trials given a predefined probability, which yields

$$\mathbb{E}\left[N_\mathrm{o}|p_\mathrm{o}\right] = 1 + \sum_{x=0}^{\infty} x p_\mathrm{o}^x (1 - p_\mathrm{o}) = 1 + \frac{p_\mathrm{o}}{1 - p_\mathrm{o}} \tag{1}$$

for the expected value of successive fixations on the same object.

Ego-motion. We propose to simulate an infant that moves with objects in its environment while holding the same object. Thus, in a similar manner as above, we also define p_s, which is the probability of sticking with the same session. For each new fixation there is a chance of $1 - p_\mathrm{s}$ that the positive pair 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. N_s plays an analogous role to choosing the frequency ego-motion in the VHE.

Speed of object manipulation. To simulate the different speed of object manipulations in humans, we distinguish two different ways of how the next view (or positive image) 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. It approximates a higher manipulation speed, but without impacting neither the marginal probability distribution of each sample nor the expected number of fixations. This approach provides the most diversity among successive views, which is expected to aid learning, but is less congruent with an infant's viewing experience.

In order to technically apply our method while controlling the parameters N_o and N_s, we use a procedure similar to Schneider et al. (2021) and extend it to account for background changes. Note that we also add inter-object switches in our augmentation set. For details about the technical implementation, see Appendix B.6.

3.5 Evaluation and training procedure

Training. To compare time-based augmentations to conventional image transformations we consider a widely adopted SOTA set of augmentations (Chen et al., 2020; Shijie et al., 2017), *i.e* crop/resize, grayscale, color jittering and horizontal flip with their default parameters, except for the crop/resize augmentation (cf. Appendix D).

In the ToyBox, VHE and 3DShapeE, the number of time-based augmentations is smaller than the possible number of conventional augmentations since we do not generate a new time-based augmentation for each training batch. For fairness, we consider the same amount of data and augmentations: 1) for the ToyBox dataset, we sample positive pairs from a pre-built dataset of conventionally augmented images; 2) for the VHE and 3DShapeE the agent collects images, augments them and stores them in a buffer of size $100,000$. 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/diversity of data related to the test-set that feeds the SSL algorithm. Thus, we also rotate the object for conventional augmentations (but do not consider it as an augmentation). When we study the combined effect of the two types of augmentations, we apply the sequence of conventional augmentations to the next image in the temporal sequence. We show in Appendix B.1 that limiting the number of conventional augmentations does not impact our analysis.

Evaluation. We evaluate the learnt representation after 480,000 steps for the VHE and 3DShapeE, 60 epochs for ToyBox and 100 epochs for CORe50. The quality of the learned representation is assessed by training a linear readout on top of the learned representation in a supervised fashion (Chen et al. (2020)). We average all results over 3 random seeds. Because of the small number of categories in the CORe50 dataset, we apply a cross-validation with 5 splits. Hyper-parameters are given in the Appendix D.

4 Results

In this section, we compare the impact of conventional data augmentations to augmentations derived from natural interactions in the context of self-supervised methods. Then, we provide an in-depth analysis of the impact of object rotations and ego-motion on the learned representations. An ablation study is available in Appendix B.

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

To validate the quality of time-based augmentations, we consider three SOTA representatives of existing paradigms in data-augmented SSL: SimCLR for contrastive learning (Chen et al., 2020), BYOL (Grill et al., 2020) for distillation-based methods and VICReg (Bardes et al., 2022) for entropy maximization methods. In Table 1, we compare the quality of the learnt representation when using conventional data-augmentations (SimCLR, BYOL, VICReg), time-based augmentations (*-TT), and their combination (C-*-TT) in our three environments and the ToyBox dataset. Time-based augmentations outperform standard data-augmentations on the 3DShapeE (except for VICReg which

Augmentations/Algorithm	3DShapeE	VHE	ToyBox	CORe50
SimCLR	0.527 ± 0.005	0.328 ± 0.002	0.313 ± 0.001	0.544 ± 0.074
SimCLR-TT	0.598 ± 0.007	0.369 ± 0.005	0.189 ± 0.009	0.449 ± 0.152
C-SimCLR-TT	$\mathbf{0.621 \pm 0.001}$	$\mathbf{0.537 \pm 0.001}$	$\mathbf{0.378 \pm 0.005}$	$\mathbf{0.610 \pm 0.090}$
BYOL	0.496 ± 0.012	0.352 ± 0.017	0.354 ± 0.007	0.552 ± 0.063
BYOL-TT	$\mathbf{0.570 \pm 0.001}$	0.369 ± 0.007	0.213 ± 0.014	0.498 ± 0.135
C-BYOL-TT	0.536 ± 0.001	$\mathbf{0.516 \pm 0.002}$	$\mathbf{0.393 \pm 0.031}$	$\mathbf{0.641 \pm 0.099}$
VICReg	0.512 ± 0.002	0.169 ± 0.006	0.381 ± 0.019	0.410 ± 0.073
VICReg-TT	0.383 ± 0.167	0.264 ± 0.012	0.158 ± 0.008	0.245 ± 0.084
C-VICReg-TT	$\mathbf{0.574 \pm 0.008}$	$\mathbf{0.435 \pm 0.013}$	$\mathbf{0.418 \pm 0.016}$	$\mathbf{0.503 \pm 0.057}$

Table 1: Top 1 accuracy $\pm$ standard deviation (over 3 seeds, CORe50 over 5 training/testing splits) under linear evaluation of our time-based augmentations (*-TT) versus previous SOTA (BYOL, SimCLR, VICReg) and their combination (C-*-TT). In 3DShapeE, **C-*-TT** use the focus change, crop/resize, color jittering and rotations. In VHE, **C-*-TT** use rotations, Ns=1 with room changes, color jittering and gray scaling. In both VHE and 3D Shape Environments, ***-TT** use rotations, Ns=1 with room changes (VHE only) and depth changes. In CORe50 we use $p_s = 0.5, p_o = 0.9$. We refer to Appendix B.4 for the ablation study that motivates these choices. In other environments, we always apply all their respective transformations. We apply all standard augmentations for the baseline.

shows a high variance) and the VHE, but not consistently in other environments. This suggests that while they are mostly crucial for shape generalization, they may be over-sensitive to object colors, like the crop/resize data-augmentation (Chen et al., 2020). In VHE, while SimCLR-TT may also be sensitive to object colors, we presume that SimCLR is even more sensitive to background details. We further study shape generalization in section 4.2 and the encoding of the background in section 4.3. Importantly, we observe that the combination of augmentations consistently outperforms conventional data-augmentations. In the next sections, we analyze the impact of rotations and speed of motion on the learnt representations.

4.2 Object manipulations including 3-D rotations support shape generalization

To assess the importance of object manipulations to generalize over object shapes, we conducted experiments in 3DShapeE. Figure 3A shows that increasing the speed of object rotations systematically increases the category recognition accuracy. Similarly in the CORe50 Environment (Figure 3D) a high manipulation speed (uniform sampling) outperforms lower manipulation speed (random walk), in particular for small number of fixations. Interestingly, we observe the inverse effect on individual object recognition accuracy (Figure 3B). We conclude that fast object rotations (in the extreme case constructing positive pairs from randomly rotated views of an object) 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 want 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 3C, we combine the flip augmentation and object rotations with other conventional augmentations. We see that when the object rotation speed is large enough ($rot(360)$), 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.

Finally, we aim to understand the relation between rotation invariance and object shape. To do so, we design a new test setup. In the *single-single setup*, we train the linear classifier on the representations of a single-view dataset that contains one view from a common orientation for each training object (back, side, ...) and assess it on the same orientation of each test object. It means that the linear classifier can not learn the invariance to rotation and does not *explicitly* need it for classification.

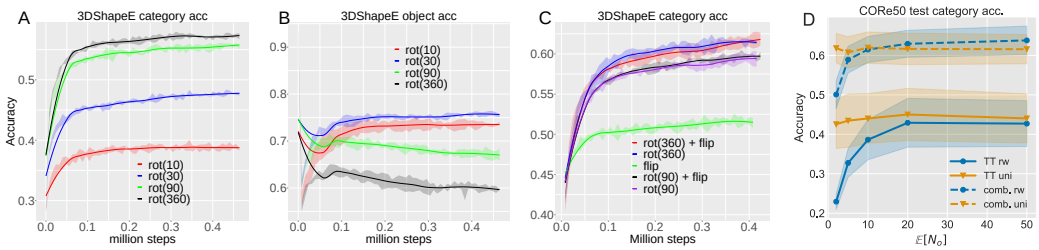

Figure 3: A. 3DShapeE test category recognition accuracy according to the maximal rotation speed S, without other augmentations. B. 3DShapeE test object recognition accuracy according to the maximal rotation speed, without other augmentations. C. 3DShapeE test category accuracy according to different combinations of flip and rotation speeds. D. Category accuracy for the CORe50 Environment when varying number of fixations ($p_s = 0.95$). The shaded area displays the $\pm$ standard deviation of test accuracy over seeds and standard error for CORe50. A-D. We used SimCLR-TT.

We consider datasets of orientations of -90° (front), -45° (¾), 0° (side) as they show the three most extreme cases in our preliminary results.

In Figure 4A, we consistently observe an ordered performance: ¾ views are better classified than side views, which are better classified than front views. We deduce that all views are not qualitatively equivalent for category recognition. While our results suggest that ¾ views are better on average, it may change according to the considered category as illustrated in Figure 4E. However, we also see in Figure 4A that the performance of C-SimCLR-TT varies across view-types. We conclude that C-SimCLR-TT, despite learning invariance to rotations, does not systematically exploit the most informative view about the category of an object to categorize its different views.

We rather hypothesize that including 3D rotation in time-based augmentations allows to encode the shape similarity in the learned representation, which results in an improved category recognition. To verify this hypothesis, we first want to check if encoding the shape similarity makes easier category recognition. To evaluate category recognition based on several views of an object (thus its whole shape), in Figure 4A, we compute a majority vote among classification results from the different single-single classifiers. It works as follows: if more views (e.g 90°, 80°, 10°) are predicted as category A than category B (e.g 0°, -90°), category A is selected. We observe a large increase of accuracy in SimCLR. We conclude that encoding the shape similarity makes easier category recognition. Second, we see a much smaller increase of accuracy for C-SimCLR-TT with the majority vote. We deduce that it better extrapolates the shape of the object from one view which turns out to crucially improve category recognition.

4.3 Moving with objects mitigates the effect of background clutter

To evaluate the impact of ego-motion while holding objects, we trained our agent in the VHE with different levels of motion. In Figure 4B, we see that increasing the frequency $1/N_s$ and the range (room) of motion considerably boosts the test category accuracy in backgrounds from training and novel houses. At the same time, we observe a decrease of the accuracy when classifying the background Figure 4C. We conclude that moving while holding objects improves the representation's invariance with respect to the background.

For the CORe50 Environment we analysed the frequency of session changes compared to the number of fixations as one of the hyperparameters of our approach. A detailed parameter study can be found in Figure 5. Qualitatively, for the validation class accuracies we observe a 2D gradient pointing from low fixations, low session changes to high fixations, high session changes. For test class accuracy, performance seems to more or less saturate at the main diagonal. This may indicate that intermediate values might actually be better than going to the extreme, but could also be partly because N_s is bounded by N_o. The classification of sessions consistently shows that session information is being disregarded when the frequency of session changes rises. However, even at the lowest setting around 50 % of the session information is still retained.

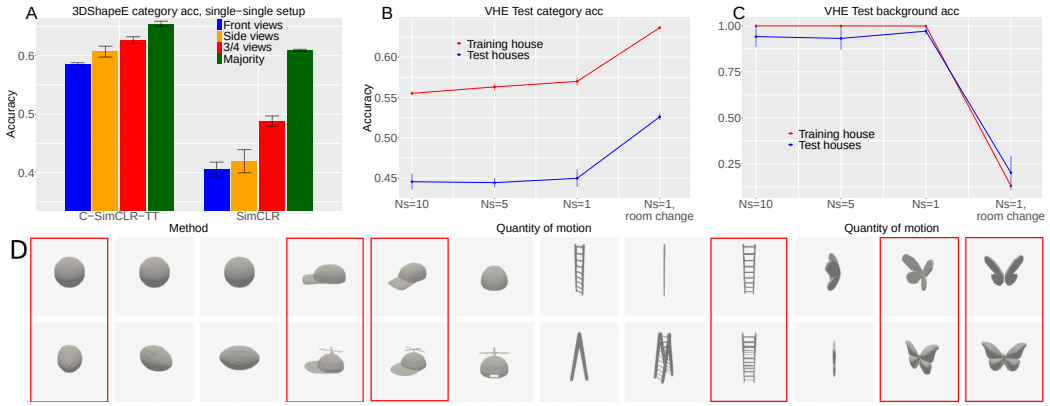

Figure 4: A. Test category accuracy with a linear classifier trained on one single orientation of objects, and tested on the same orientation of test objects. We used 5 seeds, all our conventional augmentations but only rotations ($[0, 360]$ degrees) as time-based augmentation. We apply the majority vote on datasets of six different orientations: 0°, 10°, -45°, 80°, -80°, -90°. B and C. Impact of the frequency (N_s) of body motions and whether they change rooms on the final test category accuracy (B) and the final background accuracy (C). We also used object rotations ($[0, 360]$ degrees), color jittering and grayscaling. Vertical bars indicate one standard deviation. D. Example of three views for eight objects distributed into four categories. In order from left to right for each object, we consider orientations of: 0° (side); -45° (¾); 90° (back). The red rectangles encompass visually similar views.

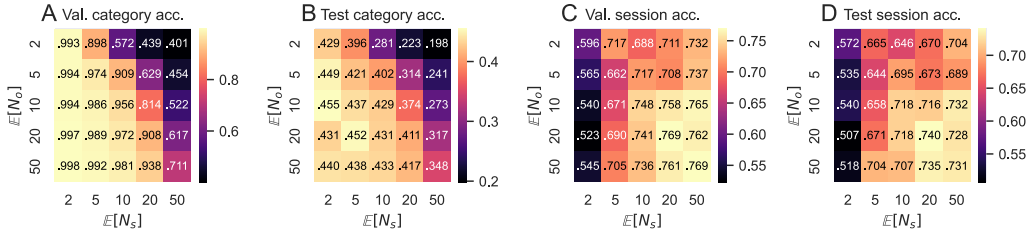

Figure 5: Mean CORe50 classification accuracy (SimCLR-TT) dependent on the number of fixations and the frequency of session switches based on 3 training/testing splits.

5 Conclusion

We investigated the impact of time-based, infant-inspired natural object interactions as a data-augmentation method on a representation learnt through SSL. 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 ability of the learned representations to generalize over object categories. Our analysis shows that 1) object rotations are crucial to build good representations of shape categories, because the representation better captures the whole shape similarity. 2) Ego-motion while holding an object so that it is seen against different backgrounds prevents the background from cluttering the learned representation.

Our work has a number of limitations. First, while we have portrayed the time-based augmentations as stemming from object *manipulations*, we did not render a hand holding the object or similar in the VHE and 3DShapeE. Second, the CORe50 and ToyBox datasets are small. Large scale experiments may give additional insights, but require a fundamental change of learning paradigm towards embodied agents that interacts with realistic environments. Finally, in this work, we only considered naive behavioral strategies for, e.g., 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 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

Removed for double blind review.

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.

Chen, X. and He, K. (2021). Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758.

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*.

Ermolov, A. and Sebe, N. (2020). Latent world models for intrinsically motivated exploration. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

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.

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.

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. In *NeurIPS 2021 Workshop on ImageNet: Past, Present, and Future*.

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

Shijie, J., Ping, W., Peiyi, J., and Siping, H. (2017). Research on data augmentation for image classification based on convolution neural networks. In *2017 Chinese automation congress (CAC)*, pages 4165–4170. IEEE.

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*.

Wang, X., Ma, T., Ainooson, J., Cha, S., Wang, X., Molla, A., and Kunda, M. (2018). The toybox dataset of egocentric visual object transformations. *arXiv preprint arXiv:1806.06034*.

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.

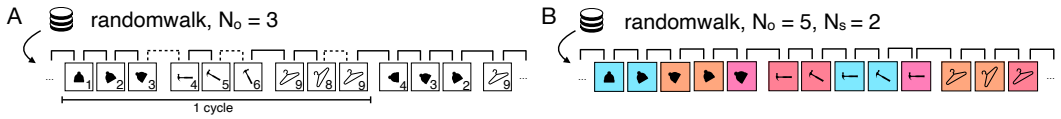

Figure 6: Dynamic sampling procedure for the CORe50 dataset. A) Buffer with deterministic $N_\mathrm{o} = 3$. Digits correspond to frame number of the video. B) Buffer with cross session sampling, colors represent different sessions.

A Complements to method

A.1 Gaze control as time-based augmentation

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.

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,F). The agent uniformly samples pan a_p and tilt a_t angles in the circle formed by an angle of 24 degrees. More precisely, we sample $a \sim U(0, 2\pi)$ and set $a_p = \cos(a)$, $a_t = \sin(a)$ This aims to highlight the possible cropping of the object.

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 within $[0.65; 2.05]$ meters. We display the largest distance of focus in Figure 2E. At object initialization or when this feature is disabled, we apply the perfect strategy of always focusing on the object. Our preliminary experiments show negative results when always applying the transformation, thus we set the fixation point outside the object with probability 0.05 and allow the gaze to move outside the object with probability 0.25.

We study the impact of gaze control in Appendix B.3.

A.2 Sampling views in the CORe50 Environment

To simulate N_o and N_s we do not directly sample from the data set, but rather build a buffer of positive pairs and iterate through it.

Building the training buffer is done in cycles, see Figure 6A. In every cycle, each different object of the data set is chosen exactly once — the order in which the objects are presented is a new random permutation during each cycle. For each sampled object, the procedure consists of three steps:

1. We randomly sample a view v_1.
2. We sample a view from the same object v_2 to form the same-object pair $(v1, v2)$ using a) *random walk*, b) *uniform. If we reach* N_s*, we sample* v_2 *from a different session (same object but with a different background).*
3. We repeat 2. to form (v_2, v_3) and continue until we reached N_o fixations.

After that, we sample a view from the next object according to the cycle (which is also a positive pair) and repeat the procedure. Thus, a cycle consists of a total of $N_\mathrm{obj} \times N_\mathrm{o}$ object views. During the fixation on one object multiple cross-session transitions can occur, see Figure 6B. Several cycles of $N_\mathrm{obj} \times N_\mathrm{o}$ views are stored in the buffer 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 dataset. In consequence, every training image on average is presented once each epoch.

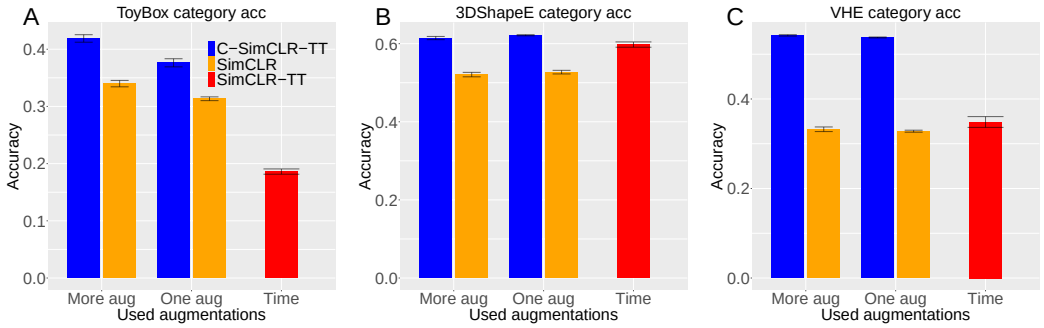

Figure 7: Test category accuracy according to whether we pre-apply conventional data-augmentations on the dataset or apply them on each minibatch (TJGCH +aug). We compare on A) the ToyBox dataset; B) 3DShapeE and C) VHE.

B Complementary analysis

B.1 The number of conventional augmentations does not matter

To fairly compare the quality of time-based augmentations with conventional ones, we restricted the number of conventional augmentations in VHE, 3DShapeE and ToyBox. Here, we aim to check whether time-based augmentations are useful even if we do not impose this restriction. In the ToyBox dataset (Figure 7A), we observe that removing this restriction improves both the baseline (SimCLR) and the combined augmentations (C-SimCLR-TT) and do not change their relative ordering. In addition, it does not significantly impact the accuracy on the 3DShapeE Figure 7B and VHE Figure 7B. We hypothesize that the increasing size of the buffer already limits over-fitting to a small set of transformations. Overall, it strengthens our analysis: time-based augmentations provide complementary and better information about the object shape in comparison to conventional data-augmentations.

B.2 The linear classifier learns about rotation invariance

Here we want to assess the importance of the linear classifier to learn the invariance over rotations in our results. We reuse the *single-single setup* used in section 4.2 and compare the results with a new *multi-single setup*. Through the *multi-single setup*, we train a linear classifier on representations gained from the usual multi-view dataset but test it on the single-view datasets.

We observe an overall decrease of performance in SimCLR when testing with the *single-single setup* (Figure 8A) in comparison to *multi-single setup* (Figure 8B). In contrast, C-SimCLR-TT (with rotations only) is more robust to the training set of the linear classifier. This suggests that the quality of the representation of SimCLR is over-estimated due to the linear classifier.

B.3 Ablation of data-augmentations in the 3D Shape Environment

In Figure 9, we test different sets of augmentations in the 3D Shape environment, progressively combined in a cumulative way. Since the combinatorial combination of all augmentations is computationally infeasible, we run a greedy forward algorithm to select the order of augmentations: at each step, we keep the augmentation with the largest increase of accuracy and stop when there is no more improvement. We find three main augmentations: rotations, crop and resize and color jittering. Adding other augmentations do not seem to have a significant large effect on the accuracy.

To better understand the impact of time-based augmentations on the representation, we also combine rotations of other time-based augmentations (R+D, R+F, R+S). While focus changes and saccades have no significant positive impact on the representation, adding depth changes achieve similar performance to the addition of crop and resize. We hypothesize it mimics the "resize" part of the crop and resize augmentation.

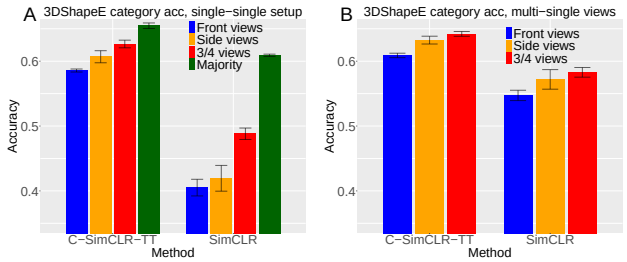

Figure 8: A. Replicated from Figure 4 for readability. Test category accuracy with a linear classifier trained on one single orientation of objects, and tested on the same orientation of test objects. B. Test category accuracy with a linear classifier trained on the multi-view dataset (several orientations of objects), and tested on test objects with one orientation. We used 5 seeds, all our conventional augmentations but only rotations ($[0, 360]$ degrees) as time-based augmentation. We apply the majority vote on datasets of six different orientations: 0°, 10°, 45°, 80°, -80°, -90°.

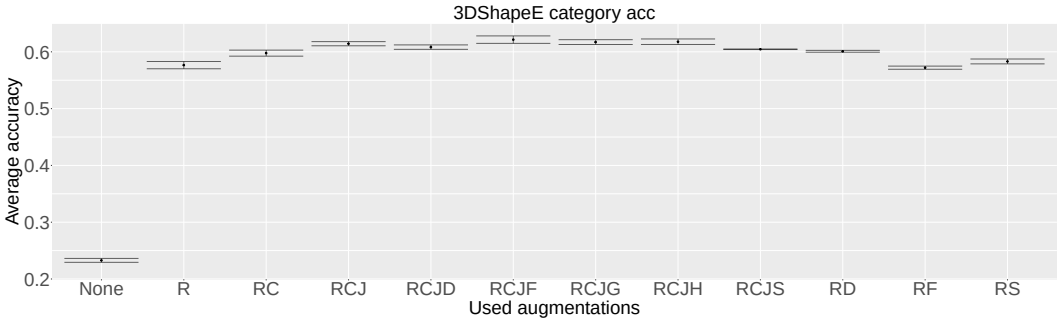

Figure 9: Ablation analysis of all data-augmentations in the 3D Shape environment. Horizontal bars indicate the minimum and maximum accuracy over 3 seeds. We note: R=Rotations; D=Depth changes; F= Focus distance; J=Color Jittering; G=Grayscale; C=Crop and Resize; H=Horizontal flip.

B.4 Ablation of data-augmentations in Virtual Home Environment

We test different sets of augmentations in the VHE, following the same procedure as in Appendix B.3. We observe that rotations, egomotion and color-based transformations are crucial for category recognition. Yet we can not conclude other augmentations are useless. It may be the redundancy with other augmentations that is harmful.

B.5 Ablation of time-based data-augmentations in the ToyBox dataset

Figure 11 shows the impact of removing time-based augmentations in the ToyBox dataset. We observe that removing augmentations based on translations (-Ho,Tr) and rotations (-Ho,R) both hurt the quality of the representation. Removing some of them only (-Ho) do not affect the performance, presumably because other time-based augmentations (rotations and translations) already provide information about the shape of the object.

B.6 Detailed analysis varying the number of fixations and session frequency on CORe50 dataset

To complement the analysis of results from the CORe50 dataset in the main manuscript (see Figure 5), we evaluate the different algorithms on 5 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 2. 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 12 shows the results in the form of box-plots.

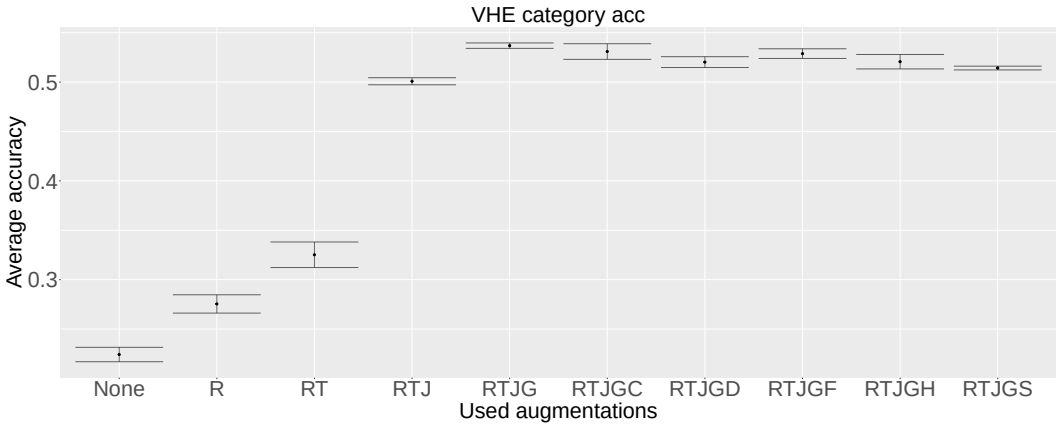

Figure 10: Ablation analysis of all data-augmentations in the Virtual Home Environment. Horizontal bars indicate the minimum and maximum accuracy over 3 seeds. We note: R=Rotations; D=Depth changes; F= Focus distance; J=Color Jittering; G=Grayscale; C=Crop and Resize; H=Horizontal flip; S=Saccades; E=Egomotion with N_s=1 and room changes.

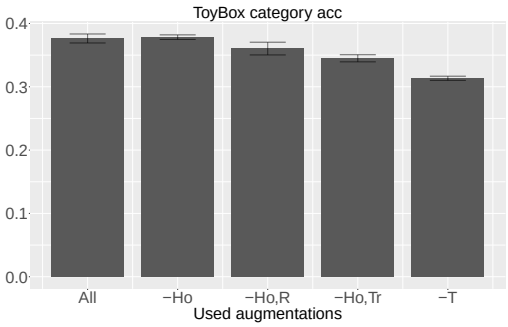

Figure 11: Ablation study on the ToyBox dataset. We apply C-SimCLR-TT but remove hodgepodge (-Ho), translations (-Tr), rotations (-R) and all time-based augmentations (-T). Experiments use the same training data and we only change how we sample the positive pair.

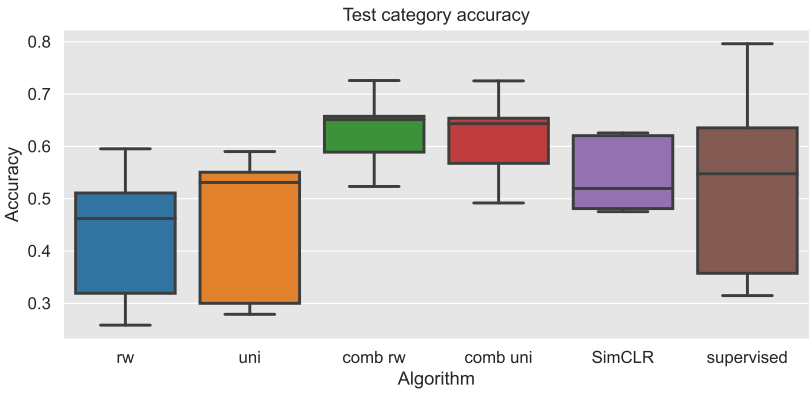

Figure 12: CORe50 Environment. Box plot of test set accuracy distribution over 5 different train/test splits after 100 epochs of training, $p_o = 0.95$, $p_s = 0.95$. Whiskers indicate min and max values.

Table 2: Category accuracy and standard error on CORe50, based on SimCLR, $p_\mathrm{s} = 0.95$. Standard error based on five different training/testing splits. Training occurred for 100 epochs, batchsize 512. Best performance is highlighted in bold.

Validation set

	$\mathbb{E}[N_\mathrm{o}]$				
algorithm	2	5	10	20	50
supervised	$.980 \pm .017$	—— " ——	—— " ——	—— " ——	—— " ——
SimCLR	$.804 \pm .011$	—— " ——	—— " ——	—— " ——	—— " ——
rw	$.430 \pm .006$	$.630 \pm .005$	$.810 \pm .007$	$.907 \pm .005$	$.939 \pm .003$
uni	$.996 \pm .001$	$\mathbf{1.000 \pm .000}$	$\mathbf{1.000 \pm .000}$	$\mathbf{1.000 \pm .000}$	$\mathbf{1.000 \pm .000}$
comb rw	$.757 \pm .005$	$.891 \pm .004$	$.940 \pm .003$	$.961 \pm .002$	$.971 \pm .001$
comb uni	$\mathbf{.997 \pm .001}$	$.998 \pm .000$	$.998 \pm .000$	$.999 \pm .000$	$.999 \pm .000$

Test set

	$\mathbb{E}[N_\mathrm{o}]$				
algorithm	2	5	10	20	50
supervised	$.530 \pm .080$	—— " ——	—— " ——	—— " ——	—— " ——
SimCLR	$.544 \pm .030$	—— " ——	—— " ——	—— " ——	—— " ——
rw	$.229 \pm .018$	$.327 \pm .033$	$.386 \pm .046$	$.429 \pm .055$	$.427 \pm .052$
uni	$.426 \pm .055$	$.435 \pm .060$	$.440 \pm .056$	$.450 \pm .059$	$.440 \pm .056$
comb rw	$.501 \pm .032$	$.589 \pm .030$	$.615 \pm .029$	$\mathbf{.629 \pm .031}$	$\mathbf{.638 \pm .032}$
comb uni	$\mathbf{.618 \pm .084}$	$\mathbf{.608 \pm .036}$	$\mathbf{.620 \pm .036}$	$.616 \pm .036$	$.616 \pm .033$

We also report means and standard errors for category (Table 3) and session accuracy (Table 4) of our randomwalk algorithm when evaluated with different values for p_s and p_o. The standard error reveals that evaluation on the validation set is fairly consistent over different training splits, while test category accuracy varies.

B.7 Influence of session changes on combined augmentations.

Since our main analysis with the CORe50 dataset aims to give a solid overview over the presented approaches we have picked a medium level of session changes as the default ($p_\mathrm{s} = 0.95$, $\mathbb{E}[N_s|p_\mathrm{s}] = 20$). 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 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_s the internal representation contains less information about the session for the randomwalk approach (C,D). The uniform method, however, shows robustness to this effect.

Table 3: Category accuracy and standard error on CORe50, based on SimCLR, randomwalk sampling procedure. Standard error based on three different training/testing splits. Training occurred for 100 epochs, batchsize 512, cf. Figure 5A, B.

Validation set

	$\mathbb{E}[N_o]$				
$\mathbb{E}[N_s]$	2	5	10	20	50
2	$.993 \pm .001$	$.994 \pm .001$	$.994 \pm .001$	$.997 \pm .000$	$.998 \pm .000$
5	$.898 \pm .009$	$.974 \pm .002$	$.986 \pm .000$	$.989 \pm .001$	$.992 \pm .001$
10	$.572 \pm .005$	$.909 \pm .005$	$.956 \pm .004$	$.972 \pm .002$	$.981 \pm .003$
20	$.439 \pm .004$	$.629 \pm .008$	$.814 \pm .010$	$.908 \pm .009$	$.939 \pm .004$
50	$.401 \pm .007$	$.454 \pm .006$	$.522 \pm .007$	$.617 \pm .003$	$.711 \pm .006$

Test set

	$\mathbb{E}[N_o]$				
$\mathbb{E}[N_s]$	2	5	10	20	50
2	$.429 \pm .069$	$.449 \pm .072$	$.455 \pm .076$	$.431 \pm .073$	$.440 \pm .073$
5	$.396 \pm .059$	$.421 \pm .072$	$.437 \pm .072$	$.452 \pm .066$	$.438 \pm .066$
10	$.281 \pm .035$	$.402 \pm .064$	$.429 \pm .077$	$.431 \pm .060$	$.433 \pm .065$
20	$.223 \pm .021$	$.314 \pm .040$	$.374 \pm .056$	$.411 \pm .063$	$.417 \pm .072$
50	$.198 \pm .025$	$.241 \pm .025$	$.273 \pm .033$	$.317 \pm .047$	$.348 \pm .054$

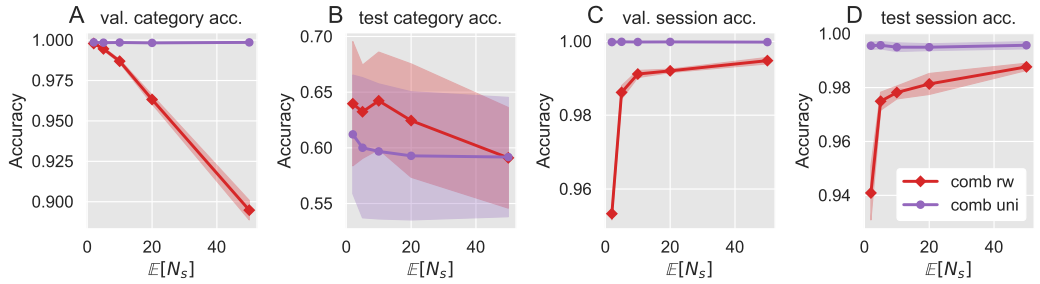

Figure 13: Analysis of the combined augmentation approach for different values of p_s, CORe50 Environment, based on SimCLR. Error bars indicate standard error based on three different training/testing splits.

C Assets

C.1 3D objects in the Virtual Home Environment

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

3DShapeE: 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

Table 4: 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 5C, D.

Validation set

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

Test set

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

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 ($< 15\%$ of the dataset). The *tree* category does not contain enough good-looking objects, so we remove the whole category.

C.2 Test datasets of the Virtual Home Environment.

Our House test set is composed of 7,740 images, from 1,548 objects distributed into 105 categories. By showcasing the category "elephant" in Figure 14 we 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 15, we display the three test houses used in Figure 4B,C. House scenes were taken from pre-provided bundles in TDW and we added the agent and objects in the scene similarly to Figure 1.

The underlying ThreeDWorld (TDW) software used is licenced under BSD 2-Clause.

C.3 ToyBox dataset

The ToyBox dataset is a collection of video-clips. We sample two frames per second from each clip, resulting in 159,627 images. Converted clips contain 7 to 50 sampled frames, which corresponds to the N_o parameter in the CORe50 environment. Two thirds of objects are used for the training set and the rest for the test set.

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 and 3D Shape Environment. The agent in the Virtual Home Environment perceives the world around it as 128×128 RGB images. These images are encoded

Figure 14: All objects of the category "elephant." used in our Virtual Home Environment.

Figure 15: 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.

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 batch normalization layer, 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 works better in this setting than standard architectures used in contrastive learning (Chen et al., 2020). We consider a temperature hyper-parameter of 0.1 for SimCLR. We use a batch size of 256 and a buffer size of 100,000. Average training time was 72 hours per run for VHE, and 20 hours for the 3DShapeE. All experiments ran on GPUs of type NVIDIA V100.

We made an hyper-parameter search of the best crop minimal size in $\{0.08, 0.2, 0.5, 0.75\}$ for the *Combined* and *SimCLR aug* experiments.

CORe50 Dataset. 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. For BYOL we use $\tau = 0.996$ and for VICReg the default error weighting as described in Bardes et al. (2022). The CORe50 dataset is licensed under CC-BY-4.0.

ToyBox dataset. We use the same hyper-parameters as for the CORe-50 experiments. However, we made an ablation study of the crop minimal size hyper-parameter to adapt to the small size and off-center position of the objects. We found a value of 0.5 to be the best in $\{0.08, 0.2, 0.5, \text{None}\}$.