Self-Supervised Learning Through Efference Copies

Franz Scherr[1*]
Huawei Technologies
franz.scherr@huawei.com

Qinghai Guo[2]
Huawei Technologies
guoqinghai@huawei.com

Timoleon Moraitis[1*]
Huawei Technologies
timoleon.moraitis@huawei.com

Abstract

Self-supervised learning (SSL) methods aim to exploit the abundance of unlabelled data for machine learning (ML), however the underlying principles are often method-specific. An SSL framework derived from biological first principles of embodied learning could unify the various SSL methods, help elucidate learning in the brain, and possibly improve ML. SSL commonly transforms each training datapoint into a pair of views, uses the knowledge of this pairing as a positive (i.e. non-contrastive) self-supervisory sign, and potentially opposes it to unrelated, (i.e. contrastive) negative examples. Here, we show that this type of self-supervision is an incomplete implementation of a concept from neuroscience, the Efference Copy (EC). Specifically, the brain also transforms the environment through efference, i.e. motor commands, however it sends to itself an EC of the full commands, i.e. more than a mere SSL sign. In addition, its action representations are likely egocentric. From such a principled foundation we formally recover and extend SSL methods such as SimCLR, BYOL, and ReLIC under a common theoretical framework, i.e. Self-supervision Through Efference Copies (S-TEC). Empirically, S-TEC restructures meaningfully the within- and between-class representations. This manifests as improvement in recent strong SSL baselines in image classification, segmentation, object detection, and in audio. These results hypothesize a testable positive influence from the brain's motor outputs onto its sensory representations.

1 Introduction

Deep Learning (DL) has drawn inspiration from Neuroscience and also offers models for understanding aspects of the brain (Richards et al., 2019). DL has been extremely successful, largely owing to labelled big datasets. However, such labelling is a costly procedure carried out by human supervisors. Fully unsupervised ML techniques do exist, however they rarely reach the performance of supervised learning (Journé et al., 2022). On the other hand, recently, a category of algorithms that are self-supervised has emerged. In self-supervised learning (SSL), the model itself generates the supervisory signal, so that human supervision is not needed, and then uses that signal for supervised learning. Recent SSL algorithms generate the supervisory signal by using pairs of inputs where it is known whether they are of the same or of a different instance, therefore self-generating positive or negative labels. Examples can be associated as being positive, e.g. based on their temporal proximity, if the input is in a sequential domain (Oord et al., 2018). Advanced SSL algorithms generate themselves the positive pairs of inputs, by augmenting the training dataset (He et al., 2020; Chen et al., 2020a; Grill et al., 2020; Caron et al., 2020; Mitrovic et al., 2021). Further improving these algorithms even has the potential to outperform supervised learning (Tomasev et al., 2022), because more information exists in the comparison of complete input pairs than in individual human-labelled examples. Therefore, improving SSL consists in devising representation-learning methods that better capture that information. However, conceptual frameworks that unify these principles of existing SSL methods, and guide towards new improved ones, are scarce (Balestriero and LeCun, 2022).

[1]Huawei Zurich Research Center, Switzerland, [2]Huawei ACS Lab, Shenzhen, China, *Corresponding author

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

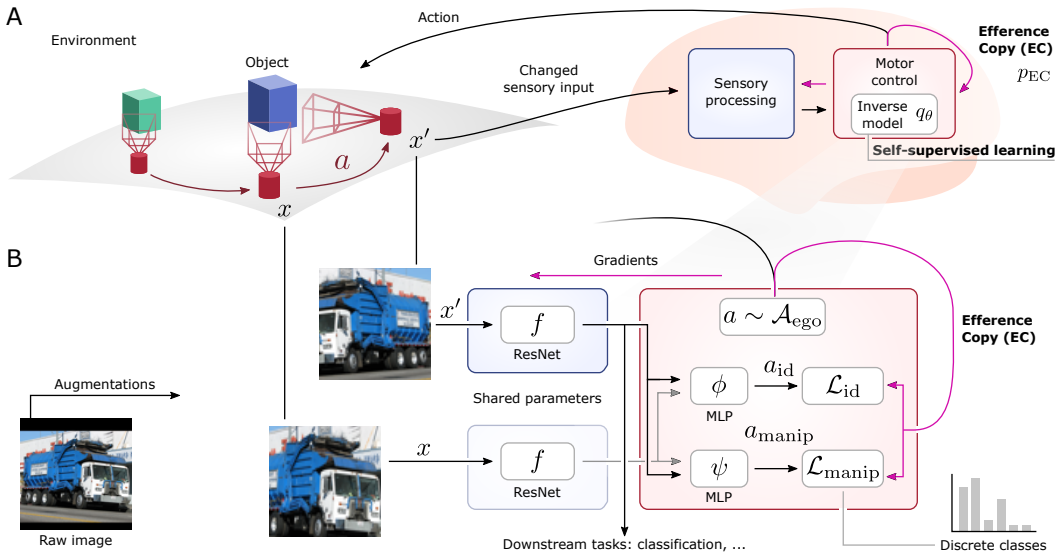

Figure 1: **Efference copy (EC). A**) Sensory-motor system: Efferent actions a (change in focus, manipulations, etc.) yield changed sensory input x'. The internal copy of the motor command, i.e. the EC, we propose, may be used as a self-supervisory signal for learning an inverse model q_θ. **B**) Abstract ML setting: The action space comprises switching and augmenting images. Sensory processing and inverse models are implemented as neural networks, trained through EC's feedback.

2 Efference copies in the central nervous system

The operation of the biological central nervous system (CNS), which appears to learn mostly without external supervision, may provide such a framework. Conversely, ML simulations within such a framework may also generate testable hypotheses for biological SSL. In the present study we take this abstract hope and formulate it as a concrete link from SSL to a specific mechanism in the CNS. We begin by observing first, that the CNS of vertebrate animals is believed to have evolved with the main purpose of performing sensory-motor control and learning, and second, that the data manipulations that augment the training examples in ML implementations of SSL can be viewed as motor actions. The search for analogies then can focus on looking for possible self-supervisory signals within biological motor control and learning.

A particularly well-suited and well-established signal in the sensory-motor system is that of the Efference Copy (EC) (von Helmholtz, 1867; McNamee and Wolpert, 2019) or Corollary Discharge (Sperry, 1950). Namely, it has been shown that when a component of the CNS addresses the body's muscles with an efferent, i.e. outgoing, motor command or action, often it also sends a copy to the CNS itself, see Fig. 1A. ECs have multiple functions and abundant supporting evidence (Kennedy et al., 2014; McNamee and Wolpert, 2019; Kilteni et al., 2020; Latash, 2021). For example, certain motor commands responsible for the locomotion of frogs are generated in the spinal cord, but are copied to the brainstem, which is responsible for motor control of the eye (von Uckermann et al., 2013). The body-movement-related disturbances to the visual field are then predicted and appropriately counteracted by eye movements that stabilize the frog's gaze. Therefore, one function of EC is to coordinate different motor controllers of the body. Another function of EC is to focus sensory processing on externally-generated and unpredicted stimuli by cancelling predictable sensations of self-generated actions. E.g., humans cannot tickle themselves effectively, because by using its ECs the CNS predicts the sensory consequence of its own action, and cancels it before it is perceived (Blakemore et al., 1998). The role of ECs in humans is actually broader and very central to motor control. Specifically, the control of bodily movements involves forward internal models that the brain maintains, i.e. models that predict the sensory inputs that result from each motor command (Kawato, 1999; McNamee and Wolpert, 2019). These forward models rely on access to motor commands to generate their predictions, and that access is provided by ECs. Importantly, motor control also involves inverse models, which map representations of targeted movement sensations to their possible actions (Rizzolatti et al., 1998; Kawato, 1999) (Fig. 1A). In addition to motor control,

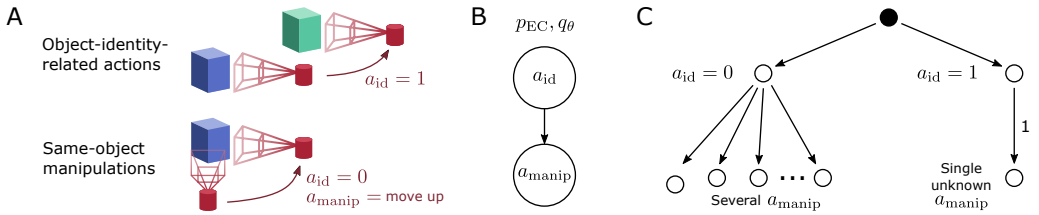

Figure 2: **Actions. A**) Categories: New objects can be brought into focus $a_{\mathrm{id}} = 1$ (e.g. saccades). Else ($a_{\mathrm{id}} = 0$), the same object can be manipulated by a_{manip} (e.g. moving). **B**) Dependency structure as a graphical model. **C**) Corresponding decision tree if a_{manip} is further assumed to be discrete.

ECs also underlie motor *learning* (Witney et al., 1999; Troyer and Doupe, 2000; Diedrichsen et al., 2003; Engert, 2013; Brownstone et al., 2015). For example, when learning an inverse model, the structures that calculate errors must access the efference.

Given the pervasive role of ECs in sensory processing, motor control, and motor learning, we hypothesize that ECs could play a key role in the learning of sensory representations too, and that it does so through the learning of inverse models. More specifically, we hypothesize that, if the EC acts as a self-supervisory learning signal, then it improves the sensory learning process, e.g. improving the later classification of input examples. Rather than physiological experiments, or biologically detailed simulations, we will test the hypothesis in an abstract ML setting. Nevertheless, we will use mechanisms that do have plausible biophysical implementations. In addition, our model could improve ML methods by providing more of the information content of paired input datapoints to SSL. That is because ECs can be rich and diverse signals, i.e. they can provide the full description of the actions that generate input pairs, and can do so for varied types of actions.

3 S-TEC: Self-supervision Through Efference Copies

3.1 Definitions and key principles

Our basic assumption is that ***(a) an EC is available***, i.e. a copy of the efferent motor commands, or actions. The essence of our strategy is to use this EC as a target label to ***(b) learn an inverse model***, mapping sensory inputs to the motor outputs that caused the changed inputs in the first place (Kawato, 1999). We assume ***(c) a hierarchical model***, e.g. a multilayer neural network. We conjecture that a model that improves on this motor-oriented task, will also improve its intermediate sensory representations as a direct consequence, which are then useful to a wider variety of sensory tasks. In our experiments, we use the representation for classification. In order to concretize the model, let $x \in \mathcal{D}$ generally denote sensory inputs. Furthermore, we define the motor commands as actions $a \in \mathcal{A}_{\text{ego}}$ that result in transformed inputs $x' = T(x, a)$, denoting with T the transformation function, see also Fig. 1A. In the following, we will simply write the EC as a probability distribution $p_{\mathrm{EC}}(a|x, x')$ to indicate the distribution of values it will assume given the sensory inputs are x before an action was taken, and are x' thereafter. We utilize this as a ground truth that the inverse model needs to predict. More formally, we denote the to-be-learned inverse sensory-motor mapping by $q_\theta(a|x, x')$ with free parameters θ that surmise synaptic weights. Learning then is the minimization of the discrepancy between the ground truth p_{EC} and our model q_θ:

$$\min_\theta \mathop{\mathbb{E}}_{\substack{x,\tilde{a}\in\mathcal{D}\times\mathcal{A}_{\text{ego}} \\ x'=T(x,\tilde{a})}} \Big[\underbrace{D_{\mathrm{KL}}(p_{\mathrm{EC}}(a|x, x'); q_\theta(a|x, x'))}_{=:\mathcal{L}\ \text{(Loss)}} \Big] . \tag{1}$$

In the above formulation, we denote with D_{KL} the Kullback-Leibler divergence, and introduce the loss function $\mathcal{L}$ that will be helpful later. The broad concept given in Eq. (1) is so far agnostic to the specific types of actions and sensory inputs. To render the matter more concrete, and to align it with the examples for sensory modalities in Section 2, we will focus on the visual sensory domain. This also facilitates the validation of our approach by ML experiments on contemporary datasets and architectures, see Section 4. We assume that only one type of sensory object is observed with each sensory input. We denote the set of possible actions as $\mathcal{A}_{\text{ego}}$. To account for the fact that the model concerns sensory-motor control in the physical world, ***(d) actions must account for two types of sensory transformations*** (see Fig. 2A), namely:

- ***(d1) Object-identity-related actions*** a_{id}. This category of action switches between sensed objects, e.g. by a saccade of the eyes, bringing entirely new objects into focus, or not. In the context of standard vision datasets that are comprised of static images, we simply exchange the currently viewed image with a randomly sampled new one. The two types of actions in this category, i.e. switching or not, are $a_{\mathrm{id}} = 1$ or $a_{\mathrm{id}} = 0$ respectively.
- ***(d2) Same-object manipulations*** a_{manip}. This category is identity-preserving, i.e. maintaining the sensed object but the observer actively manipulates it or its view, e.g.: turning the object, or moving to a closer vantage point. With static images, this kind of transformation is naturally formed by commonly used image augmentation operations. E.g. spatial transformations that crop an image with random size and random aspect ratio can simulate the movement to a different point at a closer distance whereas mirroring the image horizontally corresponds well to rotating a symmetric 3D object, see Fig. 1B. We denote an action that transforms one augmented view x into the other augmented view x' by a_{manip}.

The object-identity-related actions are of two types $a_{\mathrm{id}} \in \{0, 1\}$. Therefore, this part of the action representation is categorical. Based on this, ***(e) we model the entire action representation as categorical***, i.e. including a_{manip}. This is to follow the biological evidence that the brain maintains uniform principles throughout its organization, e.g. throughout the cortex (Douglas et al., 1989), including motor areas (Bastos et al., 2012). Moreover, there is significant evidence that this uniform organization does specifically have a categorical structure, were different actions are represented by different clusters of neurons (Graziano, 2016). Importantly, this allows learning the associated inverse model by means of a classification task, as will be introduced later.

As the overall action a is composed by two parts, i.e. $a = (a_{\mathrm{id}}, a_{\mathrm{manip}})$, we can summarize this categorical structure as a graphical model and decision tree, shown in Fig. 2B and C. Hence, given the same object continues to be in focus, i.e. $a_{\mathrm{id}} = 0$, then there exist several options for the object-manipulating action a_{manip}. In the other case, where focus is switched to a different object, i.e. $a_{\mathrm{id}} = 1$, there is no value of the object-manipulation action a_{manip} that relates x and x'. To formally represent this in the decision tree, we assign all probability to some unknown a_{manip} in that case.

So far we have not described how each class of action a_{manip}, and therefore its copy EC, is parametrized by the motor controller. Based on the fact that EC conveys to the observer the action that himself is taking, it is appropriate to ***(f) use an egocentric representation of actions*** instead of aligning the actions with an allocentric reference point, i.e. with the environment. To do so, notice that the spatial transformations introduced in (d2) are affine, thus can be represented with their associated transformation matrices. This logical parametrization allows us to conveniently compute the egocentric action that is needed to turn x into x': By multiplication of the transformation matrix that gave rise to one view from the original with the inverted transformation matrix that gave rise to the other. This highlights a difference to the allocentric representation of actions that was chosen in other work (Lee et al., 2021), where transformations were aligned to the original, allocentric reference frame (i.e. differences of scales rather than their quotient as it would emerge here).

3.2 Formalism

Through the preceding dependency structure (Fig. 2B, C), the inverse model naturally decomposes into two more specific inverse models, where one is attributed to object-identity-related actions, and the other to the same-object manipulations, to which we simply refer to as "**identity-related inverse model**" $q_\theta(a_{\mathrm{id}}|x, x')$ and "**manipulation-related inverse model**" $q_\theta(a_{\mathrm{manip}}|a_{\mathrm{id}}, x, x')$ respectively. Therefore we have that $q_\theta(a|x, x') = q_\theta(a_{\mathrm{id}}|x, x')q_\theta(a_{\mathrm{manip}}|a_{\mathrm{id}}, x, x')$. Applying the same also for the ground truth p_{EC} enables us to split the loss function into separate parts $\mathcal{L} = \mathcal{L}_{\mathrm{id}} + \mathcal{L}_{\mathrm{manip}}$ that reflect learning of the identity-related inverse model and learning of the manipulation-related inverse model correspondingly. More precisely, the loss dedicated to the identity-related inverse model is given by $\mathcal{L}_{\mathrm{id}} = D_{\mathrm{KL}}(p_{\mathrm{EC}}(a_{\mathrm{id}}|x, x'); q_\theta(a_{\mathrm{id}}|x, x'))$, while similarly, the loss dedicated to the manipulation-related inverse model is given by $\mathcal{L}_{\mathrm{manip}} = D_{\mathrm{KL}}(p_{\mathrm{EC}}(a_{\mathrm{manip}}|a_{\mathrm{id}}, x, x'); q_\theta(a_{\mathrm{manip}}|a_{\mathrm{id}}, x, x'))$, see also Appendix D for details.

In practice, we also include regularization losses $\mathcal{L}_{\mathrm{reg}}$, see Appendix C, and weight the relative importance of the loss terms by hyperparameters λ. Therefore, the loss that we consider is given by:

$$\mathcal{L} = \mathcal{L}_{\mathrm{id}} + \lambda_{\mathrm{manip}}\mathcal{L}_{\mathrm{manip}} + \lambda_{\mathrm{reg}}\mathcal{L}_{\mathrm{reg}} \ . \tag{2}$$

Instantiation of the inverse models. The specific formulation of $q_\theta(a|x, x')$ determines the loss function that will be optimized. We consider several options for the formulation of $q_\theta(a_{\text{id}}|x, x')$ that allow us to recover various contemporary approaches for SSL as we discuss in Results Section 4.1. On the other hand, for the manipulation-related inverse model, we opted for a categorical action representation that clusters similar actions as advocated in key principle (e). We implemented this by subdividing the support of each single component $a_{k,\text{manip}}$ of a_{manip} (consisting of 6 components for the affine transformation) into a number of K discrete bins. We indicate these discretized versions of the real actions with a hat $\widehat{\cdot}$ and define the probability of $a_{k,\text{manip}}$ being in bin b as:

$$q_\theta(\widehat{a}_{k,\text{manip}} = b|a_{\text{id}} = 0, x, x') = \frac{\exp\left(\psi_{k,b}(f(x), f(x'))\right)}{\sum_j \exp\left(\psi_{k,j}(f(x), f(x'))\right)} , \tag{3}$$

where we have introduced a feature extractor f (ResNets in our case, see Fig. 1) and the functions $\psi_{k,j}$ (for which we used MLPs) to express the model's belief that $a_{k,\text{manip}}$ assumes a value in discrete bin j. Note that the functions f and ψ both are learnable, but the dependence on θ is omitted for brevity. We refer to Fig. 3 for an ablation study on alternative instantiations of the manipulation-related inverse model.

4 Results

4.1 Recovering contrastive & non-contrastive SSL from the identity-related inverse model

Depending on the specific instantiation of the identity-related inverse model, we recover several common approaches for SSL using the concept of ECs. In particular, we show that based on the choice of the learned $q_\theta(a_{\text{id}}|x, x')$, we can recover from the identity-related loss either contrastive losses (i.e. instance discrimination) such as employed in SimCLR (Chen et al., 2020a), ReLIC (Mitrovic et al., 2021) or ReLICv2 (Tomasev et al., 2022), or non-contrastive approaches such as BYOL (Grill et al., 2020) (see below, and Appendices D.7.1, D.7.2, and D.7.3).

First, we consider here as an example the identity-related inverse model that gives rise to the contrastive loss of SimCLR (Chen et al., 2020a). We define this inverse model's probability of no identity-switch, i.e. $a_{\text{id}} = 0$, in the common way used for the positive view in contrastive learning (Chen et al., 2020a), for which we adopt the notation provided by Mitrovic et al. (2021):

$$q_\theta(a_{\text{id}} = 0|x, x') = \frac{\exp\left(\phi(f(x), f(x'))/\tau\right)}{\sum_{x_n \in \{x'\} \cup C} \exp\left(\phi(f(x), f(x_n))/\tau\right)} . \tag{4}$$

Here, ϕ is a function that computes a similarity between the features produced by f, i.e. it compares the intermediate sensory representations. We defined it as a scalar product between projected features: $\phi(h, h') := \sum_i g_i(h) g_i(h')$, whereas g is a multi-layer perceptron (MLP), following typical choices in the literature (see also Appendix B). The scalar τ is a temperature hyperparameter, and the set C is composed of additional candidate inputs to which x is compared to (through the denominator). Note that g is learnable also, thus depending on θ.

We assume that the EC is a perfect copy of a, hence $p_{\text{EC}}(a|x, x')$ assigns all probability to the true action a that was applied. In doing so, we obtain an upper bound of the objective (2) that we use for our S-TEC experiments (see Appendix D for the derivation). Its associated component dedicated to the identity-related inverse model is the typical contrastive learning objective:

$$\begin{aligned}\mathcal{L} \leq &- \log \frac{\exp(\phi(f(x), f(x''))/\tau)}{\sum_{x_n \in \{x'\} \cup C} \exp\left(\phi(f(x), f(x_n))/\tau\right)} \\ &- \lambda_{\text{manip}} \sum_k \log \frac{\exp\left(\psi_{k,j''}(f(x), f(x''))\right)}{\sum_j \exp\left(\psi_{k,j}(f(x), f(x''))\right)} + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}} ,\end{aligned} \tag{5}$$

where we introduced x'' to always represent an input that is related to x through $a_{\text{id}} = 0$. We use x'' also for the second term that concerns the manipulation-related inverse model, to reflect that the loss is only applied in that condition. For additional convenience we write j'' to refer to the bin in which the value of $a_{k,\text{manip}}$, relating x and x'', falls.

By integrating a confidence-estimate into the identity-related inverse model, we obtain and extend SSL methods such as ReLIC and ReLICv2. Furthermore, by defining the identity-related inverse

model as a normal distribution, we obtain non-contrastive losses such as that of BYOL. For these derivations, see Appendices D.7.2 and D.7.3. Interestingly, the loss for VICReg (Bardes et al., 2022) is of the same type as derived from S-TEC's principles for a normally-distributed identity-related inverse model.

4.2 Experimental evaluation

Table 1: Accuracies obtained with linear classification (mean and std over 5 independent runs).

Architecture	Method	CIFAR-10	CIFAR-100	STL-10
ResNet-18	SimCLR repr. (Chen et al., 2020a)	91.5 ±0.1	65.3 ±0.3	**91.5** ±0.4
	ReLIC repr. (Mitrovic et al., 2021)	91.5 ±0.2	65.6 ±0.3	**91.4** ±0.2
	MoCo v2 repr. (Chen et al., 2020c)	90.9 ±0.2	65.7 ±0.4	88.6 ±0.6
	BYOL repr. (Grill et al., 2020)	92.0 ±0.2	66.4 ±0.4	91.1 ±0.2
	S-TEC (ours)	92.0 ±0.2	66.6 ±0.3	**91.6** ±0.2
	S-TEC* (ours)	**92.6** ±0.1	**67.6** ±0.4	**91.4** ±0.2
ResNet-50	SimCLR repr. (Chen et al., 2020a)	93.2 ±0.2	70.8 ±0.2	94.0 ±0.1
	ReLIC repr. (Mitrovic et al., 2021)	93.2 ±0.2	70.8 ±0.2	93.8 ±0.1
	S-TEC (ours)	**93.9** ±0.2	71.9 ±0.4	**94.3** ±0.2
	S-TEC* (ours)	**94.0** ±0.1	**72.4** ±0.1	93.9 ±0.2

In the preceding Sections we have derived our framework that connects the concept of ECs to current methods for SSL. Here, we present the results of our experimental evaluations that aim to assess the quality of the representations that can be learned with our approach. For this purpose, we considered various image datasets, including CIFAR-10/100 (Krizhevsky et al., 2009), STL-10 (Coates et al., 2011) as well as the ImageNet ILSVRC-2012 dataset (Russakovsky et al., 2015), and compare S-TEC against several other SSL algorithms, such as SimCLR (Chen et al., 2020a), MoCo v2 (Chen et al., 2020c), BYOL (Grill et al., 2020) and ReLIC (Mitrovic et al., 2021). We follow the same procedure for all experiments, where we first performed SSL and subsequently determine the class prediction accuracy of a linear classifier that is trained on the emergent representations. During SSL the same data augmentation methods as in Chen et al. (2020a) are applied throughout, i.e. also colour augmentations, but these were not considered for training an inverse model with S-TEC, see also Appendix A for details to the augmentations applied. We adopted the ResNet v1 framework (He et al., 2016), and used specifically ResNet-18 and ResNet-50 architectures as the feature extractor f, while ϕ and ψ were generally implemented as multi-layer perceptrons (MLPs), see Fig. 1B, and Appendix B for architectural details. These networks were optimized during SSL by gradient-descent using the adaptive rate scaling of the LARS algorithm (You et al., 2017) with learning rate warmup and decay. If not otherwise stated, SSL was performed for 1,000 epochs. See Appendix C for details to the optimization with SSL and for details of training the linear classifier. For all our comparisons and ablation studies, we stress that the **overlap in the implementation is maximal**. Especially for the comparison between SimCLR and S-TEC (and between ReLIC and S-TEC*), we emphasize that only loss functions are changed and manipulation-related inverse models are added. As mentioned in Section 4.1, the identity-related inverse model can also be instantiated based on other methods for SSL, such as ReLIC (Mitrovic et al., 2021) through suitable choice of $q_\theta(a_{\text{id}}|x, x')$, see Appendix D. We denoted this specific variation with S-TEC* (i.e. target networks etc.).

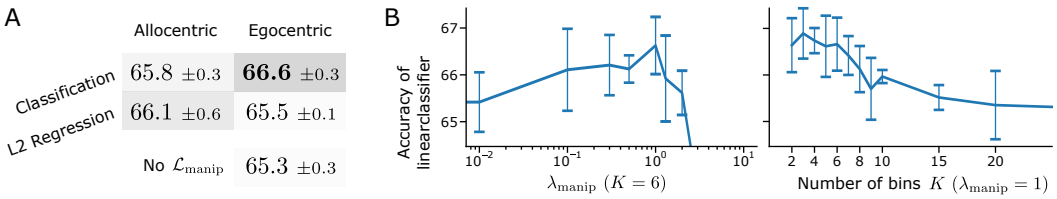

Figure 3: **Ablation study on CIFAR-100 with ResNet-18. A)** Instantiation of the manipulation-related inverse model and action representation. **B)** Hyperparameter sweep. For any setting we report mean and 95% confidence interval based on ≥5 independent runs.

CIFAR-10/100 and STL-10. We report the accuracies that linear classifiers could attain after SSL on the respective datasets in Table 1. Using 5 independent runs for each setting that we considered revealed that the manipulation-related inverse model in the case of S-TEC or S-TEC* consistently increased the accuracy of a linear classifier over the respective baseline.

Table 2: Comparing with the results of (Lee et al., 2021) on STL-10, † see Table 7 thereof.

	200 Epochs	1,000 epochs
SimSiam impl. by Lee et al. (2021)	86.32†	90.2 ±0.3
SimSiam + AugSelf (Lee et al., 2021)	86.03†	90.8 ±0.2
SimCLR repr. (Chen et al., 2020a)	86.1 ±0.2	**91.5** ±0.4
S-TEC (ours)	86.2 ±0.2	**91.6** ±0.2

We also compared our approach with the results of Lee et al. (2021), who considered a similar augmentation-aware training setting, that was mainly focused on the transferability of representations between domains. We considered the case in which their method exhibited the strongest improvement on STL-10, see Table 6 of (Lee et al., 2021) ("`crop`"), and retrained their model using the same number of epochs (1,000). We used their implementation and employed the same image augmentations as we did. Results are shown in Table 2. Conversely, we also tested our methods in a 200 epoch training budget, as originally done by Lee et al. (2021) and included the best reported performance that they obtained, see Table 7 of (Lee et al., 2021), noting that our method did not outperform in this case.

To investigate the differences between our approach and that of (Lee et al., 2021), we conducted an ablation study exchanging the egocentric action representation that we used with an allocentric one. In addition, we probed the impact of replacing action classification with L2 regression. Experiments were performed on CIFAR-100 using ResNet-18s, with Egocentric+Classification yielding **66.6**% accuracy over the next best setting Allocentric+L2 Regression with 66.1%, which was employed by Lee et al. (2021) (Fig. 3A and Appendix E). We hypothesize that classification affords the model more flexibility in its output distribution, thus it can handle uncertainty of its action prediction better.

Table 3: ImageNet results (ResNet-50).

Method (100 epoch)	Top-1 (val.)
SimCLR repr. (Chen et al., 2020a)	64.6
ReLIC repr. (Mitrovic et al., 2021)	66.2
S-TEC (ours)	64.8
S-TEC* (ours)	**66.3**
Method (300 epoch)	**Top-1 (val.)**
ReLIC repr. (Mitrovic et al., 2021)	70.0
S-TEC* (ours)	**70.2**
Method ($\geq$ 800 epoch)	**Top-1 (test)**
MoCo v2 (Chen et al., 2020c)	71.1
SwAV (Caron et al., 2020)	75.3
SimCLR (Chen et al., 2020a)	69.3
BYOL (Grill et al., 2020)	74.3
ReLIC (Mitrovic et al., 2021)	74.8
ReLICv2 (Tomasev et al., 2022)	**77.1**
VICReg (Bardes et al., 2022)	73.2

Table 4: Transfer learning on PASCAL VOC.

Method (300 epoch) - Obj. Detection (AP50)	
ReLIC repr. (Mitrovic et al., 2021)	82.3 (`test2007`)
S-TEC* (ours)	**82.5** (`test2007`)
Method (300 epoch) - Segmentation (mIoU)	
ReLIC repr. (Mitrovic et al., 2021)	69.9 (`val2012`)
S-TEC* (ours)	**70.5** (`val2012`)
Method (1000 epoch) - Segmentation (mIoU)	
BYOL (Grill et al., 2020)	76.3 (`val2012`)
ReLICv2 (Tomasev et al., 2022)	**77.9** (`val2012`)

ImageNet. We experimented with ImageNet ILSVRC-2012 (Russakovsky et al., 2015) to demonstrate that S-TEC and S-TEC* also scale. We performed training for either 100 or 300 epochs and report the results in Table 3, confirming that S-TEC is not restricted to small datasets.

Object detection and semantic segmentation. SSL aims to install generally useful representations. Thus, we considered downstream tasks beyond classification: object detection and semantic segmentation on PASCAL VOC (Everingham et al., 2010) using Faster R-CNN (Ren et al., 2015) and fully convolutional networks (Long et al., 2015), respectively, along with a ResNet-50 backbone. We initialized this backbone with the parameters that resulted from SSL on ImageNet for 300 epochs,

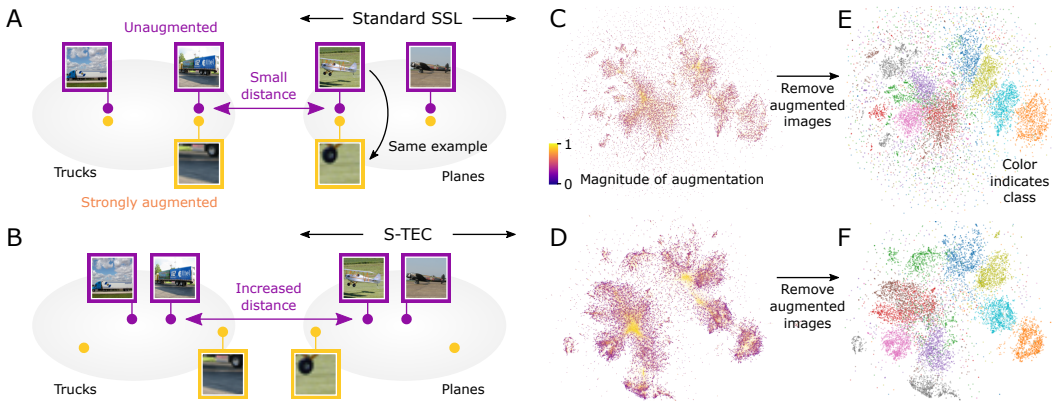

Figure 4: **Analysis. A-B**) Schematic of conjectured feature space organization: EC-unaware SSL co-locates views of an object (A), while S-TEC separates according to the level of augmentation (B). **C-D**) UMAP embeddings of images and their augmentations. Collapsed representations become separated. **E-F**) UMAP embeddings of unaugmented images, using the projection of C-D. Increased class-separation is visible.

and then fine-tuned the network on the new task (4 runs with different initializations of remaining parameters). Results are reported in Table 4, see also Appendix B and C for details.

Hyperparameter dependence and learning dynamics. To assess the dependence of our results on hyperparameters, we carried out several studies on CIFAR-100 with ResNet-18s: We performed a sweep over λ_{manip} that scales the impact of the manipulation-related loss, and a sweep over the number of bins K used in the classification for action components. Performance depends significantly on λ_{manip}, but is less affected by K, as long as there are not too many bins (i.e. $K < 10$), see Fig. 3B for results. Lastly, we also exhibit the loss dynamics and learning progress in Appendix E.3.

Audio (LibriSpeech). Finally, we also attempted to improve representation learning in the audio domain. Specifically, we considered the same data and model as introduced by Oord et al. (2018). In addition to the time-sensitive identity-related inverse model, as it emerges with CPC, we also added a time-insensitive identity-related inverse model. This allowed us to achieve **65.4**% accuracy on phoneme classification (with frozen features) as opposed to 65.1% that we obtained with CPC.

5 Analysis and intuitions

Increased information content. A consequence of learning a manipulation-related inverse model is that additional information must be expressed by the feature extractor f. Curiously, it had been shown in several other works that strong methods for SSL only perform well on downstream tasks, such as classification, if *intermediate* representations are used. E.g. Chen et al. (2020a) showed that inserting a nonlinear MLP between the feature extractor f and the loss for contrastive SSL resulted in significantly better performance on subsequent linear classification, as opposed to the control case where this MLP was missing. This effect is explained by loss of information that is not necessarily important for the contrastive SSL objective, but for downstream tasks. In fact, recent work (Chen et al., 2020b; Mitrovic et al., 2021) observed better performance if the depth of MLP was further increased. This supports the viewpoint that additional information, albeit being potentially redundant to the contrastive SSL objective, is desired for downstream tasks of interest, see also Lee et al. (2021).

Better organization of class borders. Furthermore, we conjecture that learning the additional manipulation-related inverse model using $\mathcal{L}_{\text{manip}}$ for S-TEC (see Eq. (2)) encourages the feature space to be better organized. Firstly, note that conventional (contrastive or non-contrastive) SSL promotes representations of the same object in different views to be co-located (Wang and Liu, 2021) (see Fig. 4A, e.g. truck and its tire). On the other hand, S-TEC, due to its EC-aware learning, encourages representations of different views of one object (e.g. full truck vs tire) to take different positions (truck and tire in Fig. 4B). As a result, we hypothesize, the representations of canonical,

untransformed views of the same type of object (e.g. trucks) must become more concentrated, to allow the transformed ones to spread. This must then increase the separation between clusters of untransformed objects (Fig. 4: purple arrow, A vs B). Moreover, S-TEC's separation of augmented views from unaugmented ones within an object-class allows the model to instead locate similar augmented views of different object classes. This then forms arguably semantically meaningful class-borders and transitions (Fig. 4B, truck tire and plane tire).

Experimentally, this hypothesis is supported in Fig. 4C-F, where we visualize the features computed by a ResNet-50 on images of the testing set of CIFAR-10, including also their augmentations. To do so, we computed lower-dimensional projections by the means of UMAP (McInnes et al., 2018), and colour-coded the magnitude of augmentation (defined as 1 minus the relative area of the cropped image). Comparing Fig. 4C and D, the model trained with S-TEC (Panel D) is aware of the zoom level of the augmentation and places more augmented images in similar regions, i.e. the borders, while the model trained without the full EC (SimCLR in this case) is oblivious to it (Panel C). If we compute the embeddings of the original, unaugmented, images using the preceding projection, as displayed in Fig. 4E and F, the apparent class centre distance increases for S-TEC (Panel F), due to the now missing augmented images on the border. Quantitatively, we computed for each class separately the distance between the centroid of augmented image representations and the centroid of unaugmented image representations. Averaged over all classes, we find that this distance is 10.5 for S-TEC and 0.4 for SimCLR (both latent spaces cover similar scales). This further confirms that SimCLR clusters these subsets (augmented and unaugmented images) of one class around a single centroid, whereas S-TEC separates them.

Importantly, this evidence also relates to **representational collapse**, a key issue in non-contrastive SSL (Grill et al., 2020; Bardes et al., 2022; Balestriero and LeCun, 2022). S-TEC may offer a new method for avoiding this issue, because as we have shown, it explicitly displaces representations of the same object if they correspond to different manipulations/views. We have shown how S-TEC's theoretical framework recovers non-contrastive learning and extends it with a manipulation-related inverse model, however experimental confirmation of its functionality in non-contrastive settings and its potential alleviation of representational collapse is left for future work.

6 Related work

SSL through auxiliary tasks. The idea of SSL by solving high-level queries about input manipulations was considered previously. E.g. Doersch et al. (2015); Noroozi and Favaro (2016) proposed to transform input images into patches and attempted to solve context prediction and jigsaw-puzzles respectively, while others found it useful to predict a prior rotation transformation (Gidaris et al., 2018). In contrast to such spatial prediction tasks, a different line of work by Zhang et al. (2016) discovered that colourization of black-and-white images also creates useful features for downstream tasks. Since aforementioned auxiliary tasks are orthogonal at large, prior works studied combinations and/or extensions of those (Doersch and Zisserman, 2017; Zhang et al., 2019).

Contrastive SSL. Opposing to the preceding strategies of training on (handcrafted) auxiliary tasks are algorithms that originated from the idea of mutual information (MI) maximization between the input and representations thereof. In particular, the prevalent strategy of mini-batch training rendered it practical to compare – and contrast – representations of different and related inputs, which enables the maximization of MI (Gutmann and Hyvärinen, 2010). Following this perspective, algorithms for deep networks were introduced (Oord et al., 2018), and further progress ensued, where the common blueprint for the algorithm is to transform input into pairs and to contrast those against other, unrelated ones (Henaff, 2020; He et al., 2020; Chen et al., 2020a,b, 2021). Tschannen et al. (2020) opened a discussion whether it works well due to maximization of MI, which resulted in the search for different explanations, e.g. through causal interventions (Mitrovic et al., 2021; Tomasev et al., 2022). On the other hand, the profound utility of contrastive SSL has also inspired other research that attempts to connect it to hypothesized learning mechanisms in the brain. For instance, Illing et al. (2021) show that contrastive SSL can give rise to deep representations with local learning rules.

Non-contrastive SSL. Several other studies explored non-contrastive avenues for SSL, which have gained more traction due to their attractive properties, such as not needing negative examples. In general, these approaches require the network to produce consistent representations under content-

preserving input transformations while addressing the problem of representational collapse (Grill et al., 2020; Chen and He, 2021; Zbontar et al., 2021; Bardes et al., 2022). Furthermore, Caron et al. (2020) demonstrate that enforcing consistency between cluster assignments can install potent feature extraction capabilities into a model, while also not requiring pair-wise contrasting.

Augmentation-aware self-supervision. Algorithms based on contrastive SSL typically aim for representations that are invariant to input transformations. While this seems appropriate in principle, subsequent studies have shown that this is not always favourable, as this strategy can exclude certain information from representations that could otherwise make them more useful for varying downstream tasks (Xiao et al., 2021). Based on similar arguments, Lee et al. (2021) proposed to predict differences of certain transformation parameters in addition to training on a standard SSL objective (Chen et al., 2020c,a; Chen and He, 2021) to improve the transferability of learned representations to other domains. These are promising results; however, on the main performance tests of SSL, i.e. testing the representations in the same domain as the training domain, these prior augmentation-aware approaches have not achieved the same performance advantage as compared to S-TEC, see Table 2.

The relation of our approach to prior work. Our approach generalizes methods that bring representations of paired inputs close together into a framework that also introduces semantic structure between paired inputs: by using their known relationship, i.e. transformations between them. This generalized and unified framework emerges from the concept of ECs and the feedback that it provides to create an inverse model. As a result, our approach is also augmentation-aware, but its foundation on sensory-motor principles and neuroscience instructs important elements (Section 3) that are missing from the earlier augmentation-aware approaches (Xiao et al., 2021; Lee et al., 2021), but have been studied in part by research on local learning (Illing et al., 2021). Currently, contrastive SSL is one of the dominant approaches in the literature, and our approach improves it in our tests, see Table 1 and 3. Moreover, we also confirmed that the benefits of our approach transfer to other, orthogonal improvements, such as provided by ReLIC, see same Tables. Our theoretical framework also recovers and extends non-contrastive approaches, such as BYOL (Section 4.1).

7 Conclusion

S-TEC provides a theoretical framework that is derived formally from first principles of biological sensory-motor control, unifies several important SSL approaches, and extends them through the combination of two inverse models. We showed that designing S-TEC's action representations and the inverse model in a biologically-principled way is crucial for improving performance. Performance is consistently better over several strong baselines in image classification, segmentation, and object detection. S-TEC as a framework provides a new angle from which SSL could be further improved in the future. Moreover, S-TEC is based on and constrained by established biological principles. As a result, its conclusions feed back to neuroscience, suggesting that the learning of inverse models and the availability of ECs to the nervous system may positively impact sensory skill. This hypothesis that our work here generated, is testable. It predicts that subjects exposed to a motor learning task in a novel sensory environment through active movements, would perceive the new environment better than participants that only experience passive exploration of the environment. S-TEC's biological implications could be strengthened even within the computational setting, by using optimization algorithms that are more biologically plausible than backpropagation. Multiple such options have recently been described, including within SSL (Illing et al., 2021). Adding further biophysical detail to the neural networks, such as spiking neurons, could further enhance its biological relevance.

Limitations. Even though our theoretical framework includes and extends various SSL methods, such as the very recent ReLICv2 (Tomasev et al., 2022), as well as non-contrastive SSL, e.g. BYOL (Grill et al., 2020), experimentally we have only shown improvements to the methods of Chen et al. (2020a); Mitrovic et al. (2021); Oord et al. (2018). In addition, we have employed only the basic commonly used augmentations for SSL. Improvements might be obtained by broader actions/augmentations. Another limitation is that we have explored the improvements solely for ResNet architectures (He et al., 2016). Future research might incorporate architectural improvements, and demonstrate the advantage of S-TEC in the presence of self-attention (Vaswani et al., 2017).

Potential negative societal impacts. SSL can exploit unlabelled data, and S-TEC's rich feedback from ECs improves it, thus significantly expanding also the malicious applicability of ML. One concern is of privacy, i.e. S-TEC might assist the profiling of individuals from anonymized data.

Acknowledgments and Disclosure of Funding

This work is partially supported by the Science and Technology Innovation 2030-Major Project (Brain Science and Brain-Like Intelligence Technology) under Grant 2022ZD0208700. The authors would like to thank Lukas Cavigelli, Renzo Andri, Édouard Carré, and the rest of Huawei's Von Neumann Lab, for offering compute resources.

References

Balestriero, R. and LeCun, Y. (2022). Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods. *arXiv preprint arXiv:2205.11508*.

Bardes, A., Ponce, J., and LeCun, Y. (2022). VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*.

Bastos, A. M., Usrey, W. M., Adams, R. A., Mangun, G. R., Fries, P., and Friston, K. J. (2012). Canonical microcircuits for predictive coding. *Neuron*, 76(4):695–711.

Blakemore, S.-J., Wolpert, D. M., and Frith, C. D. (1998). Central cancellation of self-produced tickle sensation. *Nature neuroscience*, 1(7):635–640.

Brownstone, R. M., Bui, T. V., and Stifani, N. (2015). Spinal circuits for motor learning. *Current opinion in neurobiology*, 33:166–173.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924.

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

Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. E. (2020b). Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255.

Chen, T., Luo, C., and Li, L. (2021). Intriguing properties of contrastive losses. *Advances in Neural Information Processing Systems*, 34.

Chen, X., Fan, H., Girshick, R., and He, K. (2020c). Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

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.

Coates, A., Ng, A., and Lee, H. (2011). An Analysis of Single Layer Networks in Unsupervised Feature Learning. In *AISTATS*. `https://cs.stanford.edu/~acoates/papers/coatesleeng_aistats_2011.pdf`.

Diedrichsen, J., Verstynen, T., Hon, A., Lehman, S. L., and Ivry, R. B. (2003). Anticipatory adjustments in the unloading task: is an efference copy necessary for learning? *Experimental Brain Research*, 148(2):272–276.

Doersch, C., Gupta, A., and Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430.

Doersch, C. and Zisserman, A. (2017). Multi-task self-supervised visual learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2051–2060.

Douglas, R. J., Martin, K. A., and Whitteridge, D. (1989). A canonical microcircuit for neocortex. *Neural computation*, 1(4):480–488.

Engert, F. (2013). Fish in the matrix: motor learning in a virtual world. *Frontiers in neural circuits*, 6:125.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338.

Gidaris, S., Singh, P., and Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. (2017). Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*.

Graziano, M. S. (2016). Ethological action maps: a paradigm shift for the motor cortex. *Trends in cognitive sciences*, 20(2):121–132.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. (2020). Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284.

Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings.

Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., and Malik, J. (2011). Semantic contours from inverse detectors. In *2011 international conference on computer vision*, pages 991–998. IEEE.

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*, pages 770–778.

Henaff, O. (2020). Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pages 4182–4192. PMLR.

Illing, B., Ventura, J., Bellec, G., and Gerstner, W. (2021). Local plasticity rules can learn deep representations using self-supervised contrastive predictions. *Advances in Neural Information Processing Systems*, 34.

Journé, A., Rodriguez, H. G., Guo, Q., and Moraitis, T. (2022). Hebbian deep learning without feedback. *arXiv preprint arXiv:2209.11883*.

Kawato, M. (1999). Internal models for motor control and trajectory planning. *Current opinion in neurobiology*, 9(6):718–727.

Kennedy, A., Wayne, G., Kaifosh, P., Alviña, K., Abbott, L., and Sawtell, N. B. (2014). A temporal basis for predicting the sensory consequences of motor commands in an electric fish. *Nature neuroscience*, 17(3):416–422.

Kilteni, K., Engeler, P., and Ehrsson, H. H. (2020). Efference copy is necessary for the attenuation of self-generated touch. *IScience*, 23(2):100843.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

Latash, M. L. (2021). Efference copy in kinesthetic perception: a copy of what is it? *Journal of neurophysiology*, 125(4):1079–1094.

Lee, H., Lee, K., Lee, K., Lee, H., and Shin, J. (2021). Improving transferability of representations via augmentation-aware self-supervision. *Advances in Neural Information Processing Systems*, 34.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

McInnes, L., Healy, J., and Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

McNamee, D. and Wolpert, D. M. (2019). Internal models in biological control. *Annual review of control, robotics, and autonomous systems*, 2:339–364.

Mitrovic, J., McWilliams, B., Walker, J. C., Buesing, L. H., and Blundell, C. (2021). Representation learning via invariant causal mechanisms. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

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.

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

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Richards, B. A., Lillicrap, T. P., Beaudoin, P., Bengio, Y., Bogacz, R., Christensen, A., Clopath, C., Costa, R. P., de Berker, A., Ganguli, S., et al. (2019). A deep learning framework for neuroscience. *Nature neuroscience*, 22(11):1761–1770.

Rizzolatti, G., Luppino, G., and Matelli, M. (1998). The organization of the cortical motor system: new concepts. *Electroencephalography and clinical neurophysiology*, 106(4):283–296.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Sperry, R. W. (1950). Neural basis of the spontaneous optokinetic response produced by visual inversion. *Journal of comparative and physiological psychology*, 43(6):482.

Tomasev, N., Bica, I., McWilliams, B., Buesing, L., Pascanu, R., Blundell, C., and Mitrovic, J. (2022). Pushing the limits of self-supervised resnets: Can we outperform supervised learning without labels on imagenet? *arXiv preprint arXiv:2201.05119*.

Troyer, T. W. and Doupe, A. J. (2000). An associational model of birdsong sensorimotor learning i. efference copy and the learning of song syllables. *Journal of neurophysiology*, 84(3):1204–1223.

Tschannen, M., Djolonga, J., Rubenstein, P. K., Gelly, S., and Lucic, M. (2020). On mutual information maximization for representation learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

von Helmholtz, H. (1867). *Handbuch der physiologischen Optik: mit 213 in den Text eingedruckten Holzschnitten und 11 Tafeln*, volume 9. Voss.

von Uckermann, G., Le Ray, D., Combes, D., Straka, H., and Simmers, J. (2013). Spinal efference copy signaling and gaze stabilization during locomotion in juvenile xenopus frogs. *Journal of Neuroscience*, 33(10):4253–4264.

Wang, F. and Liu, H. (2021). Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504.

Witney, A. G., Goodbody, S. J., and Wolpert, D. M. (1999). Predictive motor learning of temporal delays. *Journal of neurophysiology*, 82(5):2039–2048.

Xiao, T., Wang, X., Efros, A. A., and Darrell, T. (2021). What should not be contrastive in contrastive learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

You, Y., Gitman, I., and Ginsburg, B. (2017). Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. (2021). Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR.

Zhang, L., Qi, G.-J., Wang, L., and Luo, J. (2019). Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2547–2555.

Zhang, R., Isola, P., and Efros, A. A. (2016). Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] Will be in the supplemental material
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Will be in the supplemental material
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Experimental methods: Input transformations

In practice, mini-batch training was performed, hence we applied input transformations to each datapoint twice to ensure that for every x, there is always one x'' related through $a_{\mathrm{id}} = 0$, i.e. having the same underlying object identity. As a result, using a batch size of B different images will cause $2B$ images being processed at a time.

We exhibit below the complete list of data augmentation methods in the order that they were applied during SSL. Note that only *Random crop* and *Random horizontal mirroring* were considered as part of the same-object manipulations a_{manip}, see Section A.2.

1. **Random crop.**
   For each transformation we randomly extracted a patch of the image with an area sampled uniformly between 8% and 100% of the original area with an aspect ratio sampled log-uniformly between $\frac{3}{4}$ and $\frac{4}{3}$. This patch was resized to 32x32, 96x96 or 224x224 pixels for CIFAR-10/100, STL-10 or ImageNet respectively (using bilinear interpolation).
2. **Random horizontal mirroring.**
   For each transformation we mirrored the image separately with a probability of 50% horizontally.
3. **Random colour jittering.**
   With a probability of 80%, we randomly altered separately for each transformation the brightness, contrast, saturation and hue in a random order. More accurately, brightness was adjusted by multiplication of the pixel values with a factor that was uniformly sampled in $[1-u, 1+u]$ (I.e. multiplicative change in brightness). The contrast was adapted by scaling the distance of the pixels from their mean, i.e. $a(x-\mu)+\mu$, where μ was the average pixel value of the image (weighted according to red: 0.2989, green: 0.587, blue: 0.114), and a was sampled uniformly in $[1-u, 1+u]$. The saturation was adapted similarly, but in this case the mean was computed per pixel location. The change in hue was performed in HSV colour space by adding to the H channel a value sampled uniformly in $[-v, v]$ modulo 1.

   For CIFAR-10/100 u and v were set to 0.4 and 0.1 regardless of the SSL method. For STL-10 and ImageNet experiments u and v were given by 0.8 and 0.2 in the cases of SimCLR or S-TEC. In our experiments with ReLIC and S-TEC*, these values were halved.
4. **Random conversion to grayscale.**
   For each transformation the image was converted separately to grayscale with a probability of 20%. For this conversion the same weighting strategy as described above was employed.
5. **Random gaussian blur (Only for STL-10 and ImageNet).**
   With probability of 50%, we applied a Gaussian blur filter separately for each transformation. This filter had kernel edge dimensions of 10% of the image width and height (rounded to uneven edge lengths), and used a standard deviation that was sampled randomly for each transformation uniformly in [0.1, 2.0] for size 224x224 and scaled proportionally in case of other dimensions.
6. **Random solarization (Only for STL-10 and ImageNet).**
   With a probability of 20%, we also applied solarization of the image for each transformation separately. This was performed by inverting pixels with a value above 0.5 (assuming a pixel value range of 0 to 1). Here, inversion refers to a mapping $x \mapsto 1-x$.

   Note that we **excluded** the loss for the manipulation-related inverse model $\mathcal{L}_{\mathrm{manip}}$ if either x or x' had been solarized.

A.1 Object-identity-related actions a_{id}

Since we applied training in mini-batches with B different images, the action $a_{\mathrm{id}} = 1$ is simply given by x and x' corresponding to different image identities in the batch.

A.2 Same-object manipulations a_{manip}

The same-object manipulations a_{manip}, as introduced in Section 3 in the main manuscript, are only applied if the object identity of x and x' remains the same, which is the case when $a_{\mathrm{id}} = 0$.

The same-object manipulations a_{manip}, as defined in our setting, took into account only spatial operations: *Random crop* and *Random horizontal mirroring*. To represent this action, we first computed for each transformation the affine matrix M_x that generates the particular cropped view of x from the original image. More specifically, this matrix M_x transforms points on the canvas of the new view x to the points on the canvas of the original image, i.e. M_x determines the source position of the new pixels.

To compute M_x, let w_x and h_x denote the width and height of the crop in pixels as sampled from the *Random crop* operation. In addition, let l_x (t_x) be the distance of the crop's left (top) edge from the original image's left (top) edge. Additionally, let W and H denote the width and height of the original image respectively. Furthermore, let f_x be -1 if the *Random horizontal mirroring* operation dictates a mirroring and 1 if not. With these definitions M_x is defined by:

$$M_x = \begin{pmatrix} f_x \frac{w_x}{W} & 0 & \frac{w_x}{W} - 1 + 2\frac{l_x}{W} \\ 0 & \frac{h_x}{H} & 1 - \frac{h_x}{H} + 2\frac{t_x}{H} \\ 0 & 0 & 1 \end{pmatrix} =: \begin{pmatrix} m_x^{(1,1)} & m_x^{(1,2)} & m_x^{(1,3)} \\ m_x^{(2,1)} & m_x^{(2,2)} & m_x^{(2,3)} \\ m_x^{(3,1)} & m_x^{(3,2)} & m_x^{(3,3)} \end{pmatrix} . \tag{S1}$$

Since we identify with a_{manip} the action that turns x into x', we are interested in the affine transformation matrix $M_{x\to x'}$ that transforms x to x' (i.e. it computes the source location of pixels in x' on the canvas of x). It is given by:

$$M_{x\to x'} = M_{x'} M_x^{-1} . \tag{S2}$$

Finally, we identify the spatial action a_{manip} with the two top rows of this matrix:

$$a_{\text{manip}} = (m_{x\to x'}^{(1,1)}, m_{x\to x'}^{(1,2)}, m_{x\to x'}^{(1,3)}, m_{x\to x'}^{(2,1)}, m_{x\to x'}^{(2,2)}, m_{x\to x'}^{(2,3)}) \tag{S3}$$

Categorical targets. In order to classify the values of the matrix of a_{manip}, we subdivided the interval of values that can be assumed into $K = 6$ bins. For that we first define limits $\text{manip}_{\text{min}} = (-2, -2, -0.5, -2, -2, -0.5)$ and $\text{manip}_{\text{max}} = (2, 2, 0.5, 2, 2, 0.5)$, which ultimately allows us to express the discretized $\widehat{a}_{k,\text{manip}}$ of the main manuscript in Section 3.2 as:

$$\widehat{a}_{k,\text{manip}} = \max\left(\min\left(\left\lfloor \frac{a_{k,\text{manip}} - \text{manip}_{k,\text{min}}}{\text{manip}_{k,\text{max}} - \text{manip}_{k,\text{min}}} \right\rfloor, K-1\right), 0\right) . \tag{S4}$$

B Experimental methods: Architectures

The architectures for feature encoder f were residual convolutional networks as introduced by He et al. (2016) (i.e. ResNet v1). More specifically, we used ResNet-18 or ResNet-50, depending on the experiment, and used the activations after global average pooling as the output of f.

The functions ϕ, for the identity-related inverse model, and ψ, for the manipulation-related inverse model, were based on multilayer perceptrons (MLPs) with batch normalization and rectified linear activation (ReLU).

B.1 Identity-related inverse model ϕ.

The function ϕ was defined by a cosine similarity of the outputs of an MLP g:

$$\phi(a, b) = \sum_j \frac{g_j(a)}{\|g(a)\|_2} \frac{g_j(b)}{\|g(b)\|_2} , \tag{S5}$$

where the MLP g had 1 hidden layer with batch normalization and ReLU activations. Batch normalization was also used for its output (except when target networks were used in S-TEC*). The number of hidden and output units of g differed among experiments, see Table S1 for concrete dimensions.

B.2 Manipulation-related inverse model ψ.

The manipulation-related inverse model ψ was defined in the main manuscript as a function of two feature vectors. It was implemented as an MLP, applied on the concatenation of both inputs, with one hidden layer that contained 512 units with batch normalization and ReLU activation. The output of ψ was 36 dimensional in total, producing predictions for each component $a_{k,\text{manip}}$, in which of the $K = 6$ bins its value falls.

Table S1: Parameters of the MLP g per learning experiment.

Architecture for f	Parameters of g	CIFAR-10/100	STL-10	ImageNet
ResNet-18	Hidden size	512	512	-
	Output size	64	128	-
ResNet-50	Hidden size	2,048	2,048	2,048
	Output size	64	128	128

B.3 S-TEC*

We also experimented with ReLIC (Mitrovic et al., 2021), which modifies the approach by introducing target networks and an overall confidence factor $\exp(-\alpha D_{c1,c2})$, which is explained in Section D.7.3. In this case, the definition of q_θ becomes:

$$q_\theta(a_{\text{id}} = 0|x, x', \theta) = \frac{\exp(\tilde{\phi}(f(x), \tilde{f}(x'))/\tau)}{\sum_{x_n \in \{x'\} \cup C} \exp(\tilde{\phi}(f(x), \tilde{f}(x_n))/\tau)} \exp(-\alpha D_{c1,c2}) \ , \tag{S6}$$

where we used $\tilde{f}$ as a target network that follows the weights of f using an exponential moving average (Mitrovic et al., 2021) with same decay properties as in (Grill et al., 2020) using an inital decay $\tau = 0.99$. In addition, the exponential moving average was also applied to the MLP g such that $\tilde{\phi}(a, b) = \phi(a, b) = \sum_j \frac{g_j(a)}{\|g(a)\|_2} \frac{\tilde{g}_j(b)}{\|\tilde{g}(b)\|_2}$, with $\tilde{g}$ following the weights of g using an exponential moving average.

B.4 MoCo

For our implementation of MoCo v2 (Chen et al., 2020c), we used target networks $\tilde{f}$ and $\tilde{g}$ with the same exponential moving average schedule as used by (Grill et al., 2020) with an initial decay of $\tau = 0.99$. All other architectural settings were kept equal to the SimCLR setting, see Table S1. The size of the dictionary, i.e. the bank of contrastive embeddings, was set to 65k.

B.5 BYOL

For our implementation of BYOL, we followed the architectural principles as provided by Grill et al. (2020). However, for CIFAR-10/100 we used a hidden dimension of 512 for projection and predictor, as well as an output dimensionality of 64. For STL-10, we used a hidden dimension of 2048 with an output dimensionality of 128.

B.6 Object detection

For object detection, we employed Faster R-CNN (Ren et al., 2015) with a ResNet-50 backbone. In general, we followed the architectural settings of (He et al., 2020), but adopted an additional batch normalization layer not only for the box prediction head, but also for the region proposal network (RPN) just before the linear output layers.

B.7 Semantic segmentation

For semantic segmentation, we employed fully convolutional networks (Long et al., 2015) with a ResNet-50 backbone. More specifically, we followed the settings of (He et al., 2020), where we retain only convolutional layers of the ResNet, replacing stride in the last convolution block (conv_5) with a dilation of 2. After that, two 3x3 convolutions, each with batch normalization and ReLU activation, are added, followed by a 1x1 convolution for pixel-wise classification. This design yields a total stride of 16 (FCN-16s (Long et al., 2015)).

C Experimental methods: Optimization

C.1 SSL phase

We used stochastic gradient-descent with a momentum of 0.9 along with the LARS adaptive learning rate mechanism (You et al., 2017), but excluded batch normalization and bias parameters from it. We used a batch size B of 1024 for all our experiments, except for those on ImageNet, where we used a $B = 1680$. Recall that B denotes the number of different images, each of which was subject to 2 augmentations, resulting in $2B$ images processed at a time.

We employed linear scaling of the learning rate with respect to the batch size, with cosine learning rate decay and with 10 epochs of linear warmup, see Table S2 for learning rates per 256 batch size. Global weight decay was used as part of $\mathcal{L}_{\text{reg}}$ with a coefficient of 10^{-6}.

Specifically for the ResNet, we note that the last batch normalization layer in each residual block was initialized with zero scale to stabilize training (Goyal et al., 2017).

Table S2: Optimization hyperparameters per learning experiment.

Hyperparameter	CIFAR-10/100	STL-100	ImageNet
Learning rate per 256 batch size	1.0	0.3	0.3
Temperature τ	0.5	0.2	0.1
Coefficient λ_{manip} (S-TEC)	1.0	0.3	0.6

C.2 Linear classification on frozen features

For classification we trained a linear classifier on top of the frozen features using stochastic gradient descent. We trained this linear classifier alongside SSL training in the cases of CIFAR-10/100 and ImageNet, but without propagating gradients into the ResNet feature extractor f (i.e. we used `stop_gradient` for the classification loss), noting that similar results were achieved with a subsequent optimization protocol consistent with (Chen et al., 2020a).

Specifically for STL-10, we trained the linear classifier separately in a subsequent optimization procedure with stochastic gradient descent and Nesterov momentum of 0.9, using a learning rate of 0.01 per 256 batch size (we found the value of 0.175 to work best in the case of BYOL and MoCo). In this case, only random cropping and random horizontal mirroring were used as augmentation methods. This optimization program was carried out for 2,000 epochs using cosine learning rate decay and 10 epochs of warmup. For the weights of the linear classifier (excluding bias), a weight decay regularization with a coefficient of $5 \cdot 10^{-4}$ was used.

C.3 Object detection

For object detection on PASCAL VOC, we largely followed the settings of (He et al., 2020) and fine-tuned network parameters end-to-end, with training data from the `trainval2007+2012` splits, while evaluation was carried out on `test2007`. Training was performed for 24K iterations with stochastic gradient descent (using a momentum of 0.9) and a batch size of 15. The learning rate was set to 0.7, which was linearly warmed up for 1K iterations, and then multiplied by 0.1 at 18K and 22K iterations. The loss coefficient for region proposal network-related losses was set to 0.2. No weight decay was employed.

C.4 Semantic segmentation

For semantic segmentation on PASCAL VOC, we also largely followed the settings of (He et al., 2020), where training was performed on an augmented split `train_aug2012`, introduced by (Hariharan et al., 2011) for 45 epochs with stochastic gradient descent (using a momentum of 0.9) and a batch size of 16. The learning rate was set to 0.03 (0.003 for ResNet parameters initialized from SSL), which was multiplied by 0.1 at the 70% progress mark and the 90% progress mark. A weight decay of 10^{-4} was employed.

D Derivations and additional theory

D.1 Sketch of the derivation

We begin from the definition that poses the optimization of the inverse model as a minimization of the Kullback-Leibler divergence as defined in Eq. (1) of the main manuscript. Based on this and on biologically-inspired assumptions (see Section 2 of the main manuscript), the inverse model becomes a classifier of pairs of sensory inputs into actions.

Since we introduced the action as being composed of two categories, namely *Object-identity-related actions* and *Same-object manipulations* (see Fig. 2), we can decompose the loss into a sum of two losses, $\mathcal{L}_{\text{id}}$ and $\mathcal{L}_{\text{manip}}$ respectively (see Section D.9). This, in turn, allows us to learn the two associated inverse models separately.

Subsequently, we elaborate on the assumption that this model represents an embodied natural setting. From there, we show that different SSL methods emerge from S-TEC, depending on the specifics of the classifier's mathematical definition and the corresponding EC-based learning. The methods we recover include existing and proven ones, such as SimCLR (Chen et al., 2020a), BYOL (Grill et al., 2020), or ReLIC (Mitrovic et al., 2021).

D.2 Concretizing the inverse model as a classifier

The Kullback Leibler divergence loss of the inverse model (Eq. (1) of the main manuscript) is computed between (a) a probability distribution over true actions, which are copied through EC (given pairs of sensory inputs), and (b) the modelled probability distribution over actions, which is provided by the inverse model (given pairs of sensory inputs). Grounded on biological evidence (see Section 2 of the main manuscript), we assume that the probability distribution of the EC reflects a perfect copy, and that the probability distributions involved are discrete.

The inverse model estimates which actions were executed that caused the sensory inputs to change from x to x'. Based on biological evidence (see Section 2 of the main manuscript), we chose to represent this inverse model as a classifier $q_\theta(a|x, x')$ (parametrized by θ) that assigns probability to specific actions, given the inputs x before the action was executed, and the inputs x' after the action's execution.

We identified two categories of actions in Section 3 of the main manuscript, and therefore chose to view actions as being composed of two components $a = (a_{\text{id}}, a_{\text{manip}})$. As a result, two sub-inverse models (sub-classifiers) can be defined:

- the identity-related inverse model $q_\theta(a_{\text{id}}|x, x')$, and
- the manipulation-related inverse model $q_\theta(a_{\text{manip}}|a_{\text{id}}, x, x')$

by means of $q_\theta(a|x, x') = q_\theta(a_{\text{id}}|x, x')q_\theta(a_{\text{manip}}|a_{\text{id}}, x, x')$, see also Fig. 2. In turn, in order to install the function into this classifier, we defined a loss: the KL divergence in Eq. (1), which can be decomposed into a sum of two divergences, according to the aforementioned factorization. See Section D.9 for proof that the decomposition is equivalent to the original loss. This allows us not only to specify the inverse model as two separate classifiers, but also to learn them separately.

Identity-related inverse model The identity-related inverse model has to effectively solve a binary classification problem for a_{id} to identify whether the main object of the sensory inputs x is the same as the main object of the subsequent sensory input x' (in which case $a_{\text{id}} = 0$). The probability that is assigned to this event is denoted by:

$$q_\theta(a_{\text{id}} = 0|x, x', \theta) \ . \tag{S7}$$

The specific implementation of the identity-related inverse model (i.e. the classifier for a_{id}) can use a variety of criteria to determine whether two sensory inputs represent the same object identity. It is typically based on the similarity of the sensory inputs (more accurately the representations thereof). Examples of such (dis-)similarity measures include **(a) dot product**, **(b) mean squared error**, **(c) KL-divergence**, and more could be envisioned.

Manipulation-related inverse model As commented in the main manuscript, the manipulation-related inverse model has to infer which manipulations were performed, when the same underlying object remains in focus, i.e. $a_{\mathrm{id}} = 0$. Also this inverse model can be conceived in various forms, depending on which interactions and manipulations are possible for the same object. For manipulation of static images, an example is to classify the components of affine transformation matrices as we proposed in Section A.2.

D.3 Concretizing the embodied natural setting

To account for a generic and natural setting, we assume that the inverse model's parent entity, i.e. the observer or agent that performs the actions, may have contextual information from the environment in addition to the observed x and x'. More specifically, we assume that in addition to the perception of x as the main object, the agent also perceives some additional context C_{pre}. Likewise, after execution of the action a, we assume that the agent perceives x' and some additional context C. In summary, the agent perceives the set $\{x\} \cup C_{\mathrm{pre}}$ before executing the action a, and $\{x'\} \cup C$ thereafter. We further specify that in the case where $a_{\mathrm{id}} = 0$ the identity of x and x' is the same. On the other hand, in the case that the agent switches the focus to a different main object ($a_{\mathrm{id}} = 1$), then the identity of the object x remains in the broader context of the agent and is still represented in the set C in some form.

We additionally assume that the agent has prior knowledge about the physical environment's conservation laws, i.e. that one object cannot take more than one identity, and that objects do not vanish without cause. Altogether, this prior knowledge imposes a constraint on the probabilities of x having the same identity as some other x_n in the context. More accurately, we say the identity of x must be conserved in the set $\{x'\} \cup C$ after taking the action:

$$\sum_{x_n \in \{x'\} \cup C} q_\theta(a_{\mathrm{id}} = 0 | x, x_n) = 1 \ , \tag{S8}$$

which is imposed on the identity-related inverse model.

As we will show, using the entire context $\{x'\} \cup C$ during learning or only x', determines if the emerging SSL approach with S-TEC belongs to the contrastive category of methods or not.

D.4 Learning the identity-related inverse model for a_{id}

D.4.1 Using the context of an object: Contrastive SSL

When the entire context $\{x'\} \cup C$ is available during learning, utilizing the assumptions and emerging constraints from Section D.3, we arrive at two implications:

1. Learning the identity-related inverse model $q_\theta(a_{\mathrm{id}}|x, x')$ consists in maximizing the probability in Eq. (S7), if the EC dictates that x and x' share the same object identity. On the other hand, if the EC dictates that the identity of x does not match the identity of x', the probability in Eq. (S7) is to be minimized.
2. Consider specifically the case where x and x' do not share the same identity. It was assumed that the original x stays preserved in the context C in some, possibly altered, form. We denote this preserved item sharing the same identity by $x'' \in C$. Through conservation in Eq. (S8) it follows that maximizing the probability $q_\theta(a_{\mathrm{id}} = 0|x, x'')$ has as a result the minimization of the probability $q_\theta(a_{\mathrm{id}} = 0|x, x_n)$ for all other $x_n \in \{x'\} \cup C \setminus \{x''\}$ "negative" objects in the context, therefore explicit separate minimization for the negative examples is not necessary.

Furthermore, this type of learning can use directly the conservation in Eq. (S8) for mutual comparison of x with the items inside the set $\{x'\} \cup C$, typically through the use of normalization of similarities (e.g. "softmax" of similarity scores), such as how it was defined in Eq. (4) of the main manuscript.

We refer to Section D.10 for a concrete proof of the second implication, following the same idea that explicit minimization for "negative" objects is not necessary, which then connects this to the upper bound objective in Eq. (5) in the main manuscript.

D.4.2 Not using the context of an object: Non-contrastive SSL

If contextual objects (see Section D.3) are unavailable, then learning the identity-related inverse model can still be implemented by a formulation of the probability in Eq. (S7), solely on the basis of the similarity between the representations of x and x', without a comparison to the context.

That choice therefore implies a non-contrastive type of SSL. By further specifying the options of the realization of this model, we show concretely in Section D.7.2 that existing non-contrastive SSL methods emerge (Grill et al., 2020).

In this non-contrastive setting, where the task is to maximize the similarity between paired sensory representations, a trivial solution could be found, where *all* objects collapse to the same representation (Grill et al., 2020). As a result, a potentially trivial solution can occur, where *all* objects collapse to the same representation, thus maximizing the similarity of all possible representations.

This has been recognized and it has been shown that such trivial solution can be mitigated by using separate feature extractors for the two representations, and by optimizing them differently by learning in different timescales (i.e. "online networks" and "target networks") (Grill et al., 2020). We conjecture that this complexity and its drawbacks are potentially not necessary when a complete EC is employed through S-TEC, since an additional classification task involving a_{manip} must be solved that would naturally prevent such collapse, since it demands separation between representations of differently manipulated views.

D.5 Learning the manipulation-related inverse model for a_{manip}

On the other hand, we have established that in addition to the identity-related inverse model, there also exists the manipulation-related inverse model that classifies which manipulations a_{manip} were applied to an object, if the identity of x and x' remains the same.

The learning procedure of the manipulation-related inverse model depends on the specific definition of the model, as well as the type of manipulation actions a_{manip} that it models. In this work, we considered a_{manip} as being composed of the components of affine transformation matrices (see main manuscript's Section 2 for the motivation, and Sections A.2 and D.2 for details). The corresponding inverse model for a_{manip} was implemented as a classifier that predicts in which bin, i.e. class, the components of the actions fell, see Section A.2.

As a result, learning the manipulation-related inverse model consisted of training multiple classifiers for all the components of a_{manip}, in other words minimization of the cross-entropy loss between the target classes in which the components of the manipulation action a_{manip} fell and the predicted classes for these components.

D.6 Simultaneous training of two inverse models for a_{id} and a_{manip}

Ultimately we aim to train both of these inverse models simultaneously, which could be naively carried out by simply adding together the corresponding losses. However, in practice, the tasks of the two inverse models can differ in their difficulty, and thus require a different weighting to enable learning of both simultaneously. For this reason we have introduced a weighting factor λ_{manip} that scales the impact of the loss concerning the manipulation-related inverse in relation to the one corresponding to the identity-related inverse model, see Fig. 3A for a sweep over this parameter.

Furthermore, since our main goal is to achieve the best possible sensory representations installed in one model, we want to share parts of the architecture for both inverse models regarding a_{manip} and a_{id}. A direct consequence from doing so is that there may be an interaction between parts of the representation space relating to a_{manip} and other parts of the representation space relating to a_{id}, which we briefly elaborated on in Section 5 of the main manuscript.

Another conjecture, as pointed out in Section D.4.2, is that the presence of the loss relating to a_{manip} in addition to the loss relating to a_{id} could help to prevent collapse of representations, although we have not tested this hypothesis. In a similar vein, the particular point at which the two inverse models extract the respective features for their further use, and their depth, may have significant impact on the organization of representations.

D.7 Recovering prior SSL techniques from S-TEC

D.7.1 Recovering SimCLR

In case that the entire context $\{x'\} \cup C$ of objects is available and used during learning, we can obtain the NT-Xent type of loss as used in SimCLR (Chen et al., 2020a). This emerges from a specific choice for modelling the identity-related inverse model's inferred probabilities $q_\theta(a_{\text{id}} = 0|x, x_n)$, as follows.

By using the dot-product similarity between the representations of x and each example $x_n \in \{x'\} \cup C$ in the context, and then applying a "softmax" operation to convert these values into a probability, yields Eq. (4) of the main manuscript as the inverse model for a_{id}. In addition, statement (2) of Section D.4.1 is employed, which poses the learning of $q_\theta(a_{\text{id}}|x, x')$ as a maximization of $q_\theta(a_{\text{id}} = 0|x, x'')$ for a $x'' \in \{x'\} \cup C$ that shares the same identity as x. This maximization of the softmax probability for the "positive" pair is equivalent to minimizing its negative logarithm, which is, in fact, the NT-Xent loss. In Section D.10 we also show formally that NT-Xent optimizes an upper bound to the original Kullback Leibler divergence objective for learning the identity-related inverse model. Thus, we have recovered NT-Xent (Chen et al., 2020a) as a special case of our S-TEC framework, which has been the core mechanism in some of the best performing methods for SSL (Chen et al., 2020b, 2021).

D.7.2 Recovering non-contrastive SSL (BYOL)

In the case where no context is available or used during learning of the identity-related inverse model, the "BYOL" approach (Grill et al., 2020) can be recovered if the identity-related inverse sub-model of S-TEC is realized differently.

Specifically, in order to arrive at the approach of Grill et al. (2020), we begin by defining our identity-related inverse model $q_\theta(a_{\text{id}}|x, x')$ in accordance to a normal distribution, such that its optimization will result in a mean squared error loss, which is what is used in BYOL. Namely, we define:

$$q_\theta(a_{\text{id}} = 0|x, x') = (1 - \epsilon) \exp\left(-\|g(f(x)) - \tilde{g}(\tilde{f}(x'))\|_2^2\right) , \tag{S9}$$

where we have introduced $\tilde{f}$ and $\tilde{g}$ to indicate that these networks can be different from f and g but related. In Grill et al. (2020), they are related to the original network f and g via an exponential moving average (target networks). The constant ϵ denotes a small number.

We then define that the probability assigned to $a_{\text{id}} = 1$ is a small constant. Since the probability will generally not sum to 1 for these two cases, we formally introduce $a_{\text{id}} = 2$ that does not occur in practice, i.e. $p_{\text{EC}}(a_{\text{id}} = 2|x, x') = 0$:

$$q_\theta(a_{\text{id}} = 1|x, x') = \epsilon , \tag{S10}$$

$$q_\theta(a_{\text{id}} = 2|x, x') = 1 - q_\theta(a_{\text{id}} = 0|x, x') - q_\theta(a_{\text{id}} = 1|x, x') . \tag{S11}$$

Inserting these definitions into the loss of S-TEC's identity-related inverse model, $\mathcal{L}_{\text{id}} = D_{\text{KL}}(p_{\text{EC}}(a_{\text{id}}|x, x'); q_\theta(a_{\text{id}}|x, x'))$ (see Section D.9), recovers the approach of (Grill et al., 2020):

$$\begin{aligned}\mathcal{L}_{\text{id}} =& D_{\text{KL}}(p_{\text{EC}}(a_{\text{id}}|x, x'); q_\theta(a_{\text{id}}|x, x')) \\ =& \text{const} - \sum_{s=0}^{2} p_{\text{EC}}(a_{\text{id}} = s|x, x') \log q_\theta(a_{\text{id}} = s|x, x') \\ =& \text{const} + p_{\text{EC}}(a_{\text{id}}|x, x')\|g(f(x)) - \tilde{g}(\tilde{f}(x'))\|_2^2(1 - \epsilon) .\end{aligned} \tag{S12}$$

Therefore, BYOL has emerged as another special case of S-TEC.

D.7.3 Recovering ReLIC and ReLICv2

Finally, we hypothesize that the approach for ReLIC (Mitrovic et al., 2021) along with its assorted invariance penalty can also be recovered from our framework if one postulates that **both** contexts $\{x\} \cup C_{\text{pre}}$ and $\{x'\} \cup C$ (see Fig. D.3) are available and used during learning. The same principles form the basis of the more recent ReLICv2 (Tomasev et al., 2022).

In this case, the idea to arrive there is to base the classifier $q_\theta(a_{\text{id}} = 0|x, x')$ on two factors:

1. The probability of the context-aware model that was used to obtain NT-Xent, see Section D.4.1, and D.7.1,

2. An overall confidence of the identity-related inverse model that is defined based on the consistency between the contexts $\{x\} \cup C_{\text{pre}}$ and $\{x'\} \cup C$.

Consider specifically the second point that is added on top of what we considered already in the case of SimCLR in D.7.1. For the purposes of this derivation, we refer to the identity-related inverse model that emerges for obtaining SimCLR, see the definitions in D.7.1 and in D.10, as $q_\theta^{(\text{NT})}(a_{\text{id}}|x, x')$.

Furthermore, we define probability distributions $q_{c1,\theta}(x_1) = q_\theta^{(\text{NT})}(a_{\text{id}}|x, x_1)$ with $x_1 \in \{x'\} \cup C$ and $q_{c2,\theta}(x_2) = q_\theta^{(\text{NT})}(a_{\text{id}}|x'', x_2)$ with $x_2 \in \{x\} \cup C_{\text{pre}}$ in order to cross-compare probability assignments between the **same** objects in both contexts (recall that $x'' \in C$ denotes the item with the same identity as x'). The consistency between these distributions is quantified by a Kullback-Leibler divergence $D_{\text{KL}}(q_{c1,\theta}(x_1); q_{c1,\theta}(x_2)) =: D_{c1,c2}$, and is used as an overall confidence for the predictions of the inverse model in the following way:

$$\begin{aligned} q_\theta(a_{\text{id}} = 0|x, x', C_{\text{pre}}, C) &= q_\theta^{(\text{NT})}(a_{\text{id}} = 0|x, x') \exp\left(-\alpha D_{c1,c2}\right) \ , \\ q_\theta(a_{\text{id}} = 1|x, x', C_{\text{pre}}, C) &= q_\theta^{(\text{NT})}(a_{\text{id}} = 1|x, x') \exp\left(-\alpha D_{c1,c2}\right) \ , \end{aligned} \tag{S13}$$

while, similar to D.7.2, we add a $a_{\text{id}} = 2$ that does not occur in practice such as to absorb the remaining probability: $q_\theta(a_{\text{id}} = 2|\dots) = 1 - q_\theta(a_{\text{id}} = 0|\dots) - q_\theta(a_{\text{id}} = 1|\dots)$. Note that α is a hyperparameter.

It remains to substitute these definitions into $\mathcal{L}_{\text{id}}$, which yields:

$$\begin{aligned} \mathcal{L}_{\text{id}} =& D_{\text{KL}}(p_{\text{EC}}(a_{\text{id}}|x, x'); q_\theta(a_{\text{id}}|x, x')) \\ =& \text{const} - \sum_{s=0}^{2} p_{\text{EC}}(a_{\text{id}} = s|x, x') \log\left(q_\theta^{(\text{NT})}(a_{\text{id}} = s|x, x') \exp(-\alpha D_{c1,c2})\right) \\ =& \text{const} + \alpha D_{c1,c2} - D_{\text{KL}}(p_{\text{EC}}(a_{\text{id}}|x, x'); q_\theta^{(\text{NT})}(a_{\text{id}}|x, x')) \ . \end{aligned} \tag{S14}$$

Thus, we obtain the ReLIC method including its consistency loss (Mitrovic et al., 2021) by suitable definition of the inverse model in Eq. (S13).

D.8 Summary: S-TEC as a generalization of SSL methods

From first principles of sensory-motor control in Neuroscience, and the assumption that learning occurs in the physical world, we recovered prior SSL methods. However the full S-TEC model is broader, as it also includes a_{manip} in its inverse model, which is not exploited by the methods we recovered through a_{id}. In the main manuscript's Section 3, we showed that, from ECs a_{manip} is part of the same framework, and, in our experiments and analyses in the other sections of the main manuscript, we showed that it is actually useful to combine the two, if implemented according to ECs and sensory-motor principles.

Moreover, from S-TEC's framework, other powerful instantiations can be imagined. For example, we have mentioned that possibly non-contrastive approaches without a target network could become functional, by avoiding representation collapse, through a_{manip}. Further SSL concepts emerge by implementing S-TEC's elements differently, e.g. by using different technical implementations of the inverse-model's classifier.

D.9 Decomposition of the loss

In the following we show how the decomposition of the loss function $\mathcal{L}$ into the two components $\mathcal{L}_{\text{id}}$ and $\mathcal{L}_{\text{manip}}$ emerges. Starting from the definition of the loss we can expand on the definition of the

Kullback-Leibler divergence using the graphical model introduced in Fig. 2B of the main manuscript:

$$\begin{aligned}\mathcal{L} =& D_{\mathrm{KL}}(p_{\mathrm{EC}}(a|x,x');q_\theta(a|x,x'))\\ =& \sum_{s=0}^{1}\sum_{b\in\mathcal{A}_{\mathrm{manip}}} p_{\mathrm{EC}}(a_{\mathrm{id}}=s|x,x')p_{\mathrm{EC}}(a_{\mathrm{manip}}=b|a_{\mathrm{id}}=s,x,x') \qquad \text{(S15)}\\ &\cdot\log\frac{p_{\mathrm{EC}}(a_{\mathrm{id}}=s|x,x')p_{\mathrm{EC}}(a_{\mathrm{manip}}=b|a_{\mathrm{id}}=s,x,x')}{q_\theta(a_{\mathrm{id}}=s|x,x')q_\theta(a_{\mathrm{manip}}=b|a_{\mathrm{id}}=s,x,x')}\ , \qquad \text{(S16)}\end{aligned}$$

where we have introduced $\mathcal{A}_{\mathrm{manip}}$ to accommodate all possibilities that a_{manip} can realize.

This expression can be grouped differently in order to simplify:

$$\begin{aligned}=& \sum_{s=0}^{1} p_{\mathrm{EC}}(a_{\mathrm{id}}=s|x,x')\underbrace{\sum_{b\in\mathcal{A}_{\mathrm{manip}}} p_{\mathrm{EC}}(a_{\mathrm{manip}}=b|a_{\mathrm{id}}=s,x,x')}_{=1}\log\frac{p_{\mathrm{EC}}(a_{\mathrm{id}}=s|x,x')}{q_\theta(a_{\mathrm{id}}=s|x,x')}\\ &+\sum_{s=0}^{1} p_{\mathrm{EC}}(a_{\mathrm{id}}=s|x,x')\sum_{b\in\mathcal{A}_{\mathrm{manip}}} p_{\mathrm{EC}}(a_{\mathrm{manip}}=b|a_{\mathrm{id}}=s,x,x') \qquad \text{(S17)}\\ &\cdot\log\frac{p_{\mathrm{EC}}(a_{\mathrm{manip}}=b|a_{\mathrm{id}}=s,x,x')}{q_\theta(a_{\mathrm{manip}}=b|a_{\mathrm{id}}=s,x,x')}\ , \qquad \text{(S18)}\end{aligned}$$

which eventually gives rise to two separate Kullback-Leibler divergences:

$$\begin{aligned}=& D_{\mathrm{KL}}(p_{\mathrm{EC}}(a_{\mathrm{id}}|x,x');q_\theta(a_{\mathrm{id}}|x,x'))\\ &+\sum_{s=0}^{1} p_{\mathrm{EC}}(a_{\mathrm{id}}=s|x,x')D_{\mathrm{KL}}(p_{\mathrm{EC}}(a_{\mathrm{manip}}|a_{\mathrm{id}}=s,x,x');q_\theta(a_{\mathrm{manip}}|a_{\mathrm{id}}=s,x,x'))\ . \qquad \text{(S19)}\end{aligned}$$

Since $p_{\mathrm{EC}}(a_{\mathrm{manip}}|a_{\mathrm{id}}=1,x,x')$ and $q_\theta(a_{\mathrm{manip}}|a_{\mathrm{id}}=1,x,x')$ are fixed and 1 for the same, formally introduced, unknown a_{manip}, we obtain:

$$\begin{aligned}\mathcal{L} =& D_{\mathrm{KL}}(p_{\mathrm{EC}}(a_{\mathrm{id}}|x,x');q_\theta(a_{\mathrm{id}}|x,x'))\\ &+p_{\mathrm{EC}}(a_{\mathrm{id}}=0|x,x')D_{\mathrm{KL}}(p_{\mathrm{EC}}(a_{\mathrm{manip}}|a_{\mathrm{id}}=0,x,x');q_\theta(a_{\mathrm{manip}}|a_{\mathrm{id}}=0,x,x'))\\ =:& \mathcal{L}_{\mathrm{id}}+\mathcal{L}_{\mathrm{manip}}\ . \qquad \text{(S20)}\end{aligned}$$

D.10 Instance discrimination as an upper bound to the identity-related inverse model loss

In the following, we provide proof that instance discrimination is an upper bound to $\mathcal{L}_{\mathrm{id}}$ as introduced in Section 4.1 of the main manuscript. Overall, the idea to achieve this, is to recognize that training the identity-related inverse model to always identify the correct positive example (i.e. related through $a_{\mathrm{id}}=0$) will at the same time allow this model to predict the case $a_{\mathrm{id}}=1$.

We begin by the definition of the loss for learning the identity-related inverse model:

$$\begin{aligned}\mathcal{L}_{\mathrm{id}} =& D_{\mathrm{KL}}(p_{\mathrm{EC}}(a_{\mathrm{id}}|x,x');q_\theta(a_{\mathrm{id}}|x,x'))\\ =& \text{const}-\sum_{s=0}^{1} p_{\mathrm{EC}}(a_{\mathrm{id}}=s|x,x')\log q_\theta(a_{\mathrm{id}}=s|x,x')\ . \qquad \text{(S21)}\end{aligned}$$

Since there are only two possibilities for $a_{\mathrm{id}}\in\{0,1\}$ it follows that:

$$\begin{aligned}=& \text{const}-p_{\mathrm{EC}}(a_{\mathrm{id}}=0|x,x')\log q_\theta(a_{\mathrm{id}}=0|x,x')\\ &-p_{\mathrm{EC}}(a_{\mathrm{id}}=1|x,x')\log(1-q_\theta(a_{\mathrm{id}}=0|x,x'))\ . \qquad \text{(S22)}\end{aligned}$$

From the definition of $q_\theta(a_{\mathrm{id}}=0|x,x')$ in Eq. (4) in the main manuscript, we have that $\sum_{x_n\in C} q_\theta(a_{\mathrm{id}}=0|x,x_n)=1-q_\theta(a_{\mathrm{id}}=0|x,x')$, which further implies that we can take any $x''\in C$ and obtain the inequality $q_\theta(a_{\mathrm{id}}=0|x,x'')\leq 1-q_\theta(a_{\mathrm{id}}=0|x,x')$.

Inserting this inequality into Eq. (S22), and due to the monotony of the logarithm, we obtain:

$$\begin{aligned}\mathcal{L}_{\mathrm{id}} \leq& \text{const}-p_{\mathrm{EC}}(a_{\mathrm{id}}=0|x,x')\log q_\theta(a_{\mathrm{id}}=0|x,x')\\ &-p_{\mathrm{EC}}(a_{\mathrm{id}}=1|x,x')\log(q_\theta(a_{\mathrm{id}}=0|x,x''))\ , \qquad \text{(S23)}\end{aligned}$$

Finally, if on the other hand p_{EC} represents a perfect copy of a_{id}, then we can define x'' to always represent the example that forms a positive pair with x' (i.e. through $a_{\mathrm{id}} = 0$) and obtain Eq. (5) in the main manuscript, which is the instance discrimination task of (Chen et al., 2020a):

$$\mathcal{L}_{\mathrm{id}} \leq \mathrm{const} - \log(a_{\mathrm{id}} = 0 | x, x'') \ , \tag{S24}$$

where $\mathrm{const} = 0$ is a result of 0 entropy in $p_{\mathrm{EC}}(a_{\mathrm{id}}|x, x')$.

E Additional results

E.1 Visual depiction of the distribution of accuracies

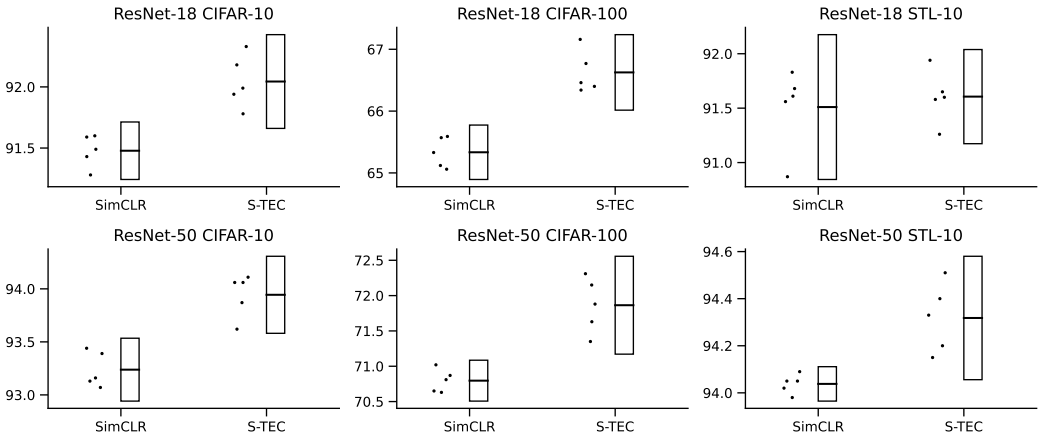

Figure S1: Visualization of performance distribution for SimCLR and S-TEC. In each setting 5 independent runs were conducted, resulting in different performances (points). These data points were used to obtain an estimated 95% confidence interval (bar).

In addition to the results presented in Table 1 to 3 in the main manuscript, Fig. S1 depicts the distribution of performance values for SimCLR and S-TEC visually.

E.2 Ablation

We also performed an ablation study for ResNet-18s trained on CIFAR-100 to more specifically assess which components helped to improve the representations, as measured by linear classification accuracy, the most. Subject to this study were two key mechanisms that were previously introduced: **categorical** and **egocentric** action representation.

Categorical action representation. The modelling of actions in a categorical manner induces a softmax cross entropy loss for the optimization of the manipulation-related inverse model, which we refer to as "Classification". Alternatively, actions and the predictions thereof can be represented in their continuous form, which gives rise to a standard L2 regression loss, which we denote as "L2 Regression", pointing out that this strategy was employed by Lee et al. (2021).

Egocentric action representation. On the other hand, our manipulation-related inverse model was trained to predict the actions that would be required to move from one view into the other based on its own perspective. This egocentric viewpoint is in contrast to the allocentric approach chosen in (Lee et al., 2021), where differences in the view are predicted based on the original image: i.e. for random cropping, differences in the cropping scale and differences of the crop's borders from the top and the left of the original image are predicted.

We tested the possible combinations of the choices for action representation (optimizing separately $\lambda_{\mathrm{manip}} \in \{0.1, 0.2, 0.5, 1.0, 2.0\}$) and report the average performance of 5 independent runs each in Fig. 3A of the main manuscript. These results confirm that categorical and egocentric action representations perform best as evaluated on linear classification accuracy.

E.3 Optimization progress

For insight in the optimization dynamics, we provide learning curves for runs of SimCLR and S-TEC on the datasets of CIFAR-10 (see Fig. S2 and S5), CIFAR-100 (see Fig. S3 and S6) as well as loss curves in the case of STL-10 (see Fig. S4 and S7). All of the provided curves were obtained using 5 independent runs for each scenario that was considered. We report the averages of these as bold curves, which were additionally processes using a moving average filter. Unprocessed individual metrics are shown as thin transparent lines.

We report in each scenarios the following metrics:

1. Loss of the manipulation-related objective (only for S-TEC),
2. accuracy of the manipulation-related inverse model (only for S-TEC), which is defined as the average accuracy that this inverse models picks the correct action clusters (measured on the training set),
3. loss of the identity-related inverse model, and
4. accuracy of the identity-related inverse model, which is reported as the fraction of positive views x and x'' being correctly identified, see also D.10.

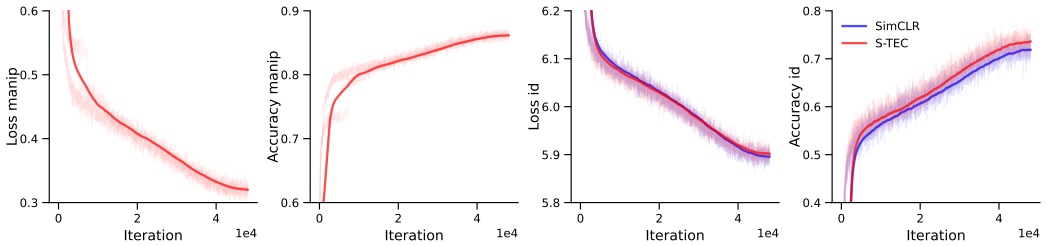

Figure S2: **ResNet-18 on CIFAR-10**: Progression of loss functions corresponding to the manipulation- and identity-related inverse model along with the accuracy of the respective task (training-set).

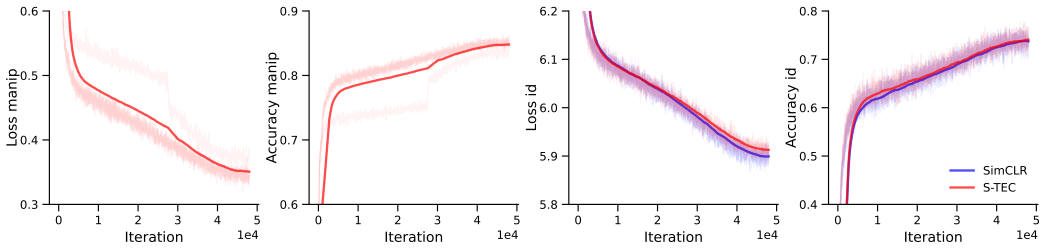

Figure S3: **ResNet-18 on CIFAR-100**: Progression of loss functions corresponding to the manipulation- and identity-related inverse model along with the accuracy of the respective task (training-set).

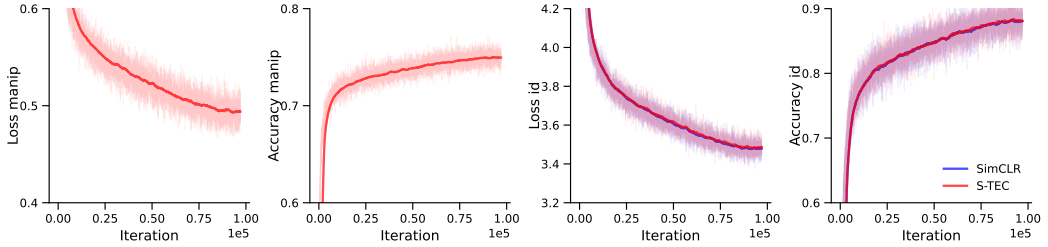

Figure S4: **ResNet-18 on STL-10**: Progression of loss functions corresponding to the manipulation- and identity-related inverse model along with the accuracy of the respective task (training-set).

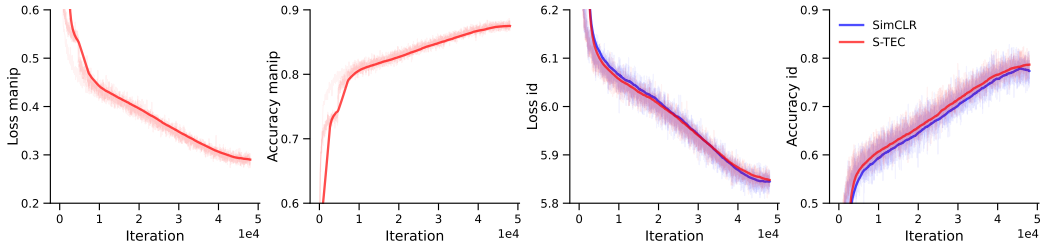

Figure S5: **ResNet-50 on CIFAR-10**: Progression of loss functions corresponding to the manipulation- and identity-related inverse model along with the accuracy of the respective task (training-set).

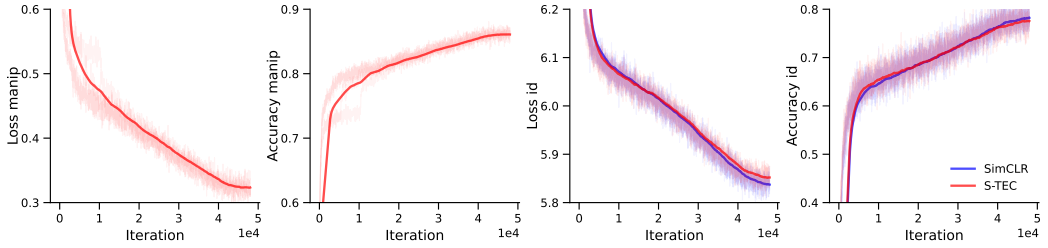

Figure S6: **ResNet-50 on CIFAR-100**: Progression of loss functions corresponding to the manipulation- and identity-related inverse model along with the accuracy of the respective task (training-set).

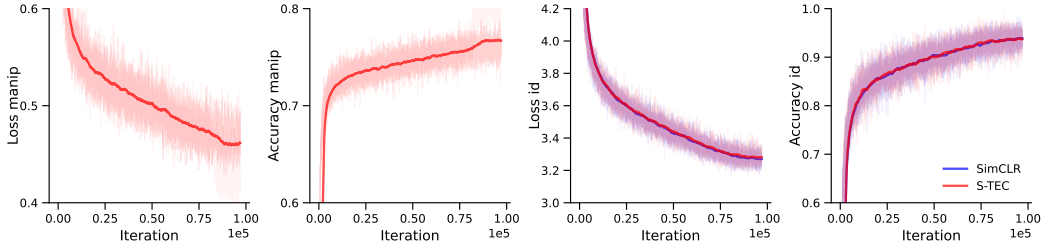

Figure S7: **ResNet-50 on STL-10**: Progression of loss functions corresponding to the manipulation- and identity-related inverse model along with the accuracy of the respective task (training-set).