Appendix to: 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

Contents

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

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

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 3.1 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 $\text{manip}_{\min} = (-2, -2, -0.5, -2, -2, -0.5)$ and $\text{manip}_{\max} = (2, 2, 0.5, 2, 2, 0.5)$, which ultimately allows us to express the discretized $\widehat{a}_{k,\text{manip}}$ of the main manuscript in Section 3.2 as:

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

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, $\mathcal{L}_{\mathrm{id}}$ and $\mathcal{L}_{\mathrm{manip}}$ 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 3.1 of the main manuscript, and therefore chose to view actions as being composed of two components $a = (a_{\mathrm{id}}, a_{\mathrm{manip}})$. As a result, two sub-inverse models (sub-classifiers) can be defined:

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

by means of $q_\theta(a|x, x') = q_\theta(a_{\mathrm{id}}|x, x')q_\theta(a_{\mathrm{manip}}|a_{\mathrm{id}}, x, x')$, 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 $a_{\mathrm{id}} = 0$). The probability that is assigned to this event is denoted by:

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

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 $\{x\} \cup C_{\text{pre}}$ before executing the action a, and $\{x'\} \cup C$ thereafter. We further specify that in the case where $a_{\text{id}} = 0$ 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 ($a_{\text{id}} = 1$), 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:

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

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 $q_\theta(a_{\text{id}} | x, x')$ 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 $q_\theta(a_{\text{id}} = 0 | x, x'')$ has as a result the minimization of the probability $q_\theta(a_{\text{id}} = 0 | x, x_n)$ 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. 3B 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 $q_\theta^{(\mathrm{NT})}(a_{\mathrm{id}}|x, x')$.

Furthermore, we define probability distributions $q_{c1,\theta}(x_1) = q_\theta^{(\mathrm{NT})}(a_{\mathrm{id}}|x, x_1)$ with $x_1 \in \{x'\} \cup C$ and $q_{c2,\theta}(x_2) = q_\theta^{(\mathrm{NT})}(a_{\mathrm{id}}|x'', x_2)$ with $x_2 \in \{x\} \cup C_{\mathrm{pre}}$ 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 $D_{\mathrm{KL}}(q_{c1,\theta}(x_1); q_{c1,\theta}(x_2)) =: D_{c1,c2}$, and is used as an overall confidence for the predictions of the inverse model in the following way:

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

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

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

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

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 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 $\mathcal{L}_{\mathrm{id}}$ and $\mathcal{L}_{\mathrm{manip}}$ 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:

$$= \sum_{s=0}^{1} p_{\mathrm{EC}}(a_{\mathrm{id}} = s|x, x') \underbrace{\sum_{b \in \mathcal{A}_{\mathrm{manip}}} p_{\mathrm{EC}}(a_{\mathrm{manip}} = b|a_{\mathrm{id}} = s, x, x')}_{=1} \log \frac{p_{\mathrm{EC}}(a_{\mathrm{id}} = s|x, x')}{q_\theta(a_{\mathrm{id}} = s|x, x')}$$

$$+ \sum_{s=0}^{1} p_{\mathrm{EC}}(a_{\mathrm{id}} = s|x, x') \sum_{b \in \mathcal{A}_{\mathrm{manip}}} p_{\mathrm{EC}}(a_{\mathrm{manip}} = b|a_{\mathrm{id}} = s, x, x') \tag{S17}$$

$$\cdot \log \frac{p_{\mathrm{EC}}(a_{\mathrm{manip}} = b|a_{\mathrm{id}} = s, x, x')}{q_\theta(a_{\mathrm{manip}} = b|a_{\mathrm{id}} = s, x, x')} , \tag{S18}$$

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

$$= D_{\mathrm{KL}}(p_{\mathrm{EC}}(a_{\mathrm{id}}|x, x'); q_\theta(a_{\mathrm{id}}|x, x'))$$

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

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:

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

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 3.2 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:

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

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

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

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:

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

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 $p_{\mathrm{EC}}(a_{\mathrm{id}}|x, x')$.

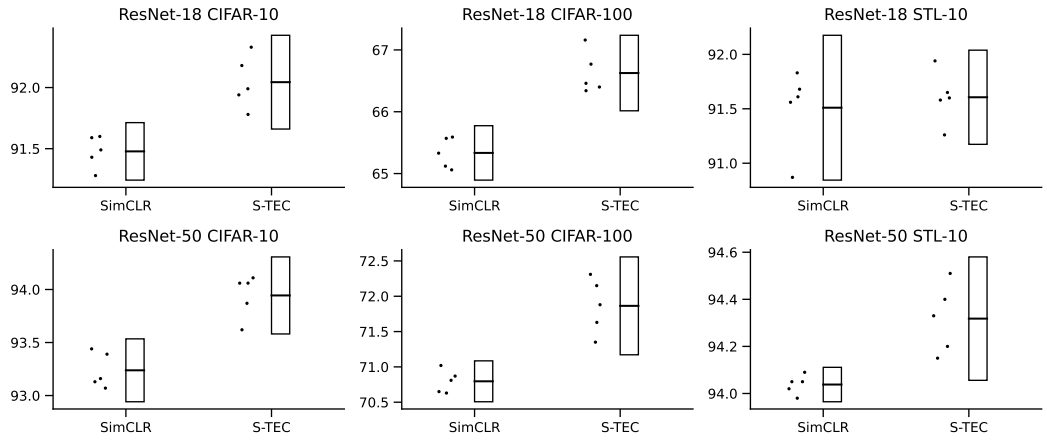

Figure S1: Visualization of performance distribution for SimCLR and S-TEC. In each setting 5 independent runs were conducted, resulting in different performances (points). These data points were used to obtain an estimated 95% confidence interval (bar).

E Additional results

E.1 Visual depiction of the distribution of accuracies

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 $\lambda_{\text{manip}} \in \{0.1, 0.2, 0.5, 1.0, 2.0\}$) 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 scenario 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.

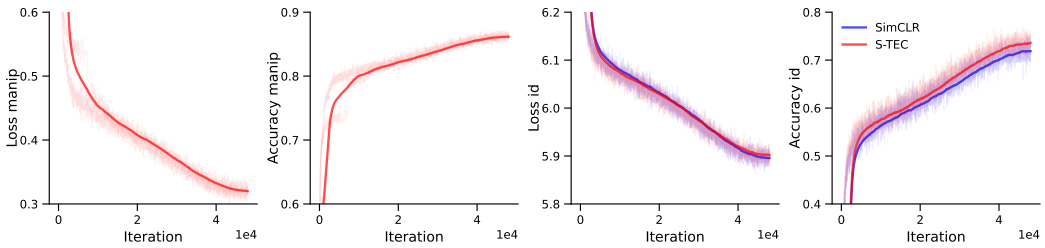

Figure S2: **ResNet-18 on CIFAR-10**: Progression of loss functions corresponding to the manipulation- and identity-related inverse model along with the accuracy of the respective task (training-set).

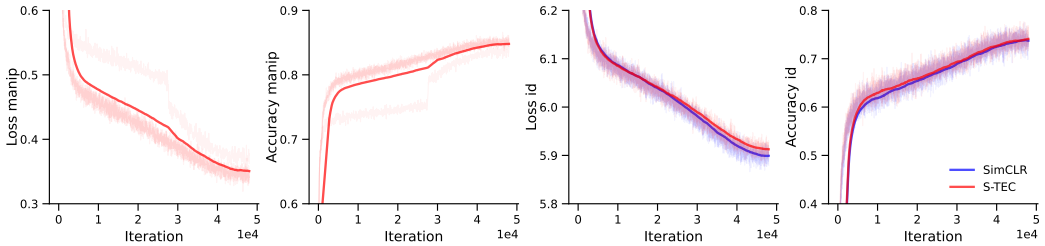

Figure S3: **ResNet-18 on CIFAR-100**: Progression of loss functions corresponding to the manipulation- and identity-related inverse model along with the accuracy of the respective task (training-set).

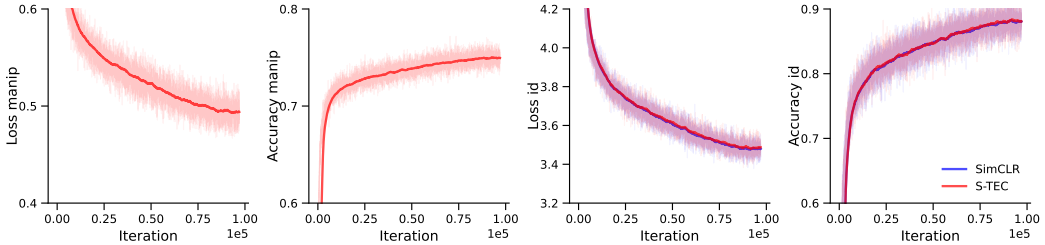

Figure S4: **ResNet-18 on STL-10**: Progression of loss functions corresponding to the manipulation- and identity-related inverse model along with the accuracy of the respective task (training-set).

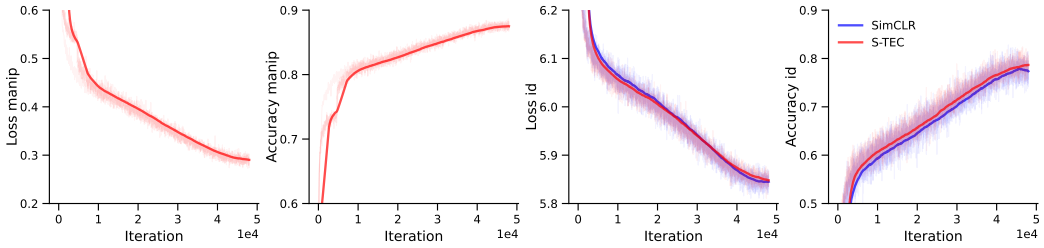

Figure S5: **ResNet-50 on CIFAR-10**: Progression of loss functions corresponding to the manipulation- and identity-related inverse model along with the accuracy of the respective task (training-set).

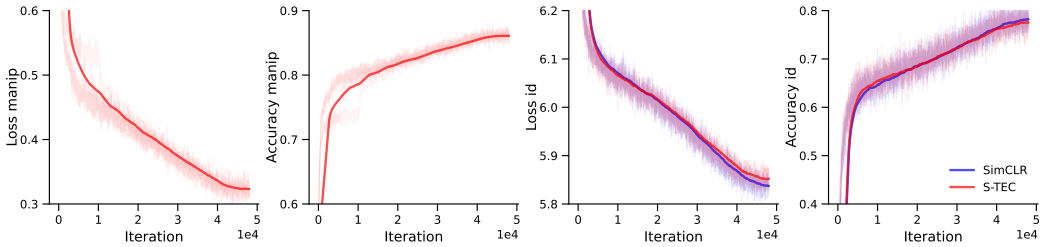

Figure S6: **ResNet-50 on CIFAR-100**: Progression of loss functions corresponding to the manipulation- and identity-related inverse model along with the accuracy of the respective task (training-set).

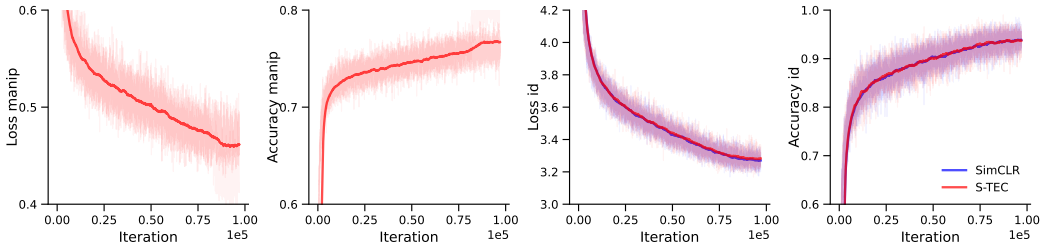

Figure S7: **ResNet-50 on STL-10**: Progression of loss functions corresponding to the manipulation- and identity-related inverse model along with the accuracy of the respective task (training-set).

References

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*.

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*.

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.

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.

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.

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.

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.

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*.

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