Self-Supervised Learning Through Efference Copies

Anonymous Author(s)
Affiliation
Address
email

Abstract

State-of-the-art (SOTA) machine learning (ML) without labels is based on self-supervised learning (SSL). SSL advances are highly demanded given the scarcity of labelled data and the cost of human supervision. Biology can offer inspiration for advanced SSL models that could also bring insights into learning in the brain. SSL commonly transforms each training datapoint into a pair of views, uses the knowledge of this pairing as a positive self-supervisory sign, and potentially opposes it to unrelated, negative examples. Here, first we show that this type of self-supervision is a concrete – albeit constrained – implementation of a concept from neuroscience: the Efference Copy (EC). Specifically, like SSL, the brain also transforms the environment through efference, i.e. outgoing motor commands, however it sends to itself an EC of the full commands, i.e. more than a mere sign. Second, we implement this insight as an SSL framework of Self-supervision Through Efference Copies (S-TEC), and we show empirically that S-TEC improves recent strong SSL baselines on image classification. Third, we hypothesize and provide preliminary evidence that S-TEC's improvement to the representation is by placing towards the border of each class heavily transformed data specifically, resulting in better separation between the classes' supports and more structured inter-class space. Overall, S-TEC conceptually generalizes the SSL framework, and hypothesizes a testable positive influence from motor outputs onto sensory representations of the brain.

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. As a result, supervised learning excludes the bulk of available data, and limits the tasks accessible to ML. Fully unsupervised ML techniques do exist, however they rarely reach the performance of supervised learning. 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 input datapoints 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, for example based on their temporal proximity, if the input is in a sequential domain (Oord et al., 2018). The most 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 can additionally alleviate the need for the costly human supervision of DL. Moreover, it even has the potential to outperform supervised learning (Tomasev et al., 2022), because more information exists in the comparison of complete input pairs as opposed to human-labelled individual

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

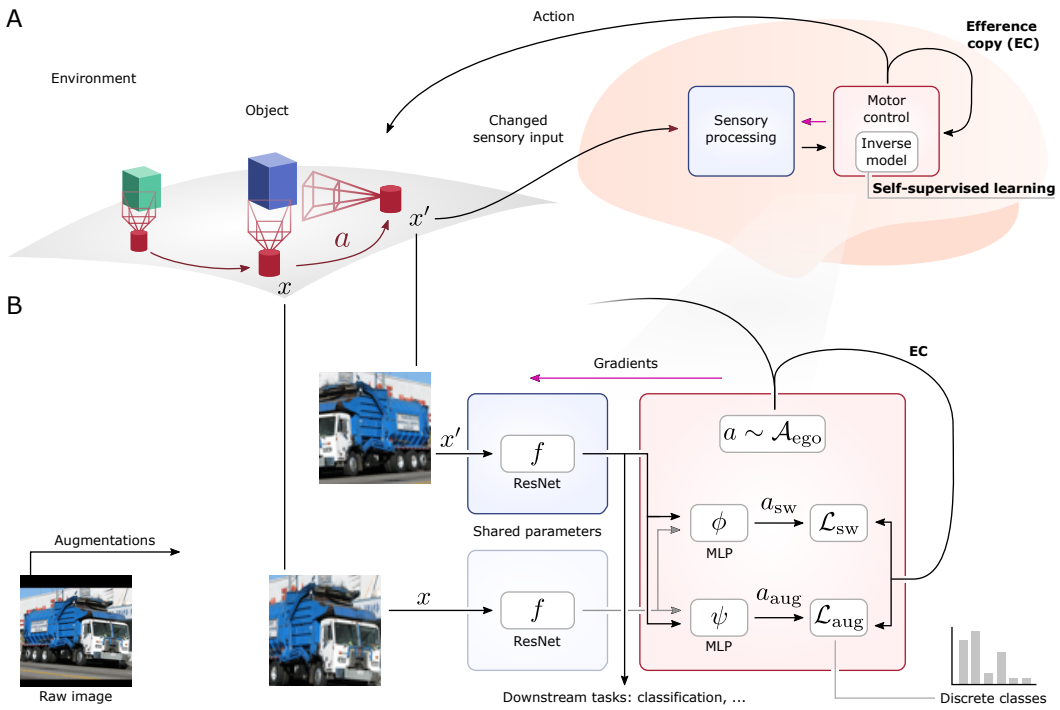

Figure 1: Efference copies. A) Schematic of sensory-motor interaction. B) Instantiation of A in an abstract ML setting.

examples. Therefore, improving SSL consists in devising representation learning methods that better capture that information.

2 Efference copies in the central nervous system

Inspiration for such improvements may be found in the operation of the biological central nervous system (CNS), which appears to learn mostly without external supervision. 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 Figure 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. For example, 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

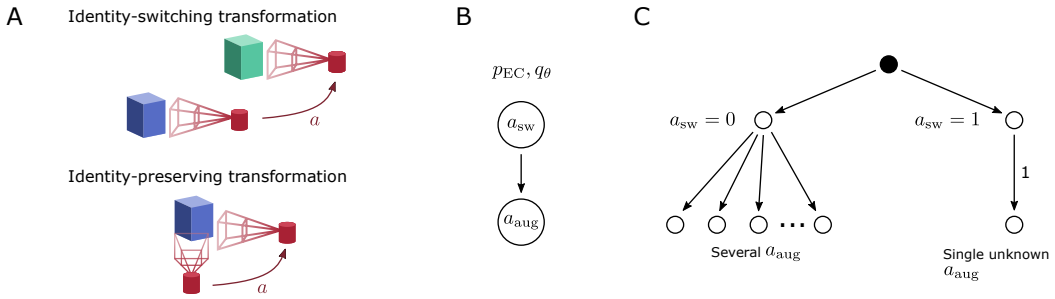

Figure 2: Actions. A) Distinguished action categories. B) Graphical model. C) Decision tree.

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) (Figure 1A). In addition to motor control, 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 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 Figure 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 equation (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 Figure 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 $a_{\text{id}} = 1$ or $a_{\text{id}} = 0$ 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. Examples could include turning the object, or moving to a closer vantage point. Interestingly, working 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 Figure 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 $a_{\text{id}} \in \{0, 1\}$. 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 as well, i.e. $a = (a_{\text{id}}, a_{\text{manip}})$, we can summarise this categorical structure as a graphical model and decision tree, depicted in Figure 2B and C. These can be summarised as follows. Given that the same object continues to be in focus, i.e. $a_{\text{id}} = 0$, then there exist several possibilities for the object-manipulating action a_{manip}. In the other case, where a switch in the viewed identity happens, i.e. $a_{\text{id}} = 1$, 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 establish this here, notice that the spatial transformations introduced in (d2) are affine, and thus can be represented with their associated transformation matrices. This logical parametrization also 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 inverse of the transformation matrix that was used to transform the original to the other view. 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).

Through the preceding dependency structure (Figure 2B, C), the inverse model naturally decomposes into two more specific inverse models, where one is attributed object-identity-related actions, and the other to the same-object manipulations, to which we simply refer to as "identity-related inverse model" and "manipulation-related inverse model" respectively. As will be shown later in our derivation, the identity-related inverse model can be learned through the instance discrimination task that is regularly employed by contemporary SSL (Mitrovic et al., 2021), hence establishing a link between EC and SSL, while learning the manipulation-related inverse model provides a link to augmentation-aware training (Lee et al., 2021).

3.2 Formalism.

In the following, we show how the loss function that we used for our experiments emerges from the idea in equation (1). Firstly, we decompose the inverse model q_θ according to the graphical model, as introduced in Figure 2B, where we obtain $q_\theta(a|x,x') = q_\theta(a_{\mathrm{id}}|x,x')q_\theta(a_{\mathrm{manip}}|a_{\mathrm{id}},x,x')$. Doing 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 respectively. 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:

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

Identity-related inverse model. The identity-related inverse-model needs to decide whether or not two views depict the same object or not. A similar problem is the instance discrimination task, commonly employed in current SSL (Chen et al., 2020a) that requires the model to identify which instances are the same given different augmentations. It turns out that an optimization of the instance discrimination task has the minimization of $\mathcal{L}_{\mathrm{id}}$ as a result, see Appendix D. Therefore chose to use this approach to optimize our identity-related inverse model, noting that other SSL methods, such as the approach of Grill et al. (2020) could be used as well, see Appendix D.

Hence, following Chen et al. (2020a), we define the probability of no switch in identity, i.e. $a_{\mathrm{id}} = 0$, by adopting the notation provided by Mitrovic et al. (2021):

$$q_\theta(a_{\mathrm{id}} = 0|x,x',\theta) = \frac{\exp\left(\phi(f(x),f(x'))/\tau\right)}{\sum_{x_n\in\{x'\}\cup C}\exp\left(\phi(f(x),f(x_n))/\tau\right)} \ , \tag{3}$$

where f is a feature extractor (e.g. ResNet-based), and ϕ compares the outputs of f, i.e. the intermediate sensory representations, see Figure 1B. 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 all functions f, ϕ are learnable, but for brevity the dependence on θ is omitted.

Manipulation-related inverse model. The manipulation-related inverse model is only subject to learning when $a_{\mathrm{id}} = 0$. In that case, it is tasked to figure out which augmentations need to be applied to x in order to turn it into x'. As mentioned above, we considered to group similar actions into clusters. We implemented this by subdividing the support of each single component $a_{k,\mathrm{manip}}$ of a_{manip} (consisting of $K = 6$ components for the affine transformation) into a number of discrete bins, and we indicate these discretized versions of the real actions with a hat $\widehat{\cdot}$. This affords a similar formulation for the probability of $\widehat{a}_{k,\mathrm{manip}}$ as above:

$$q_\theta(\widehat{a}_{k,\mathrm{manip}}|a_{\mathrm{id}} = 0,x,x',\theta) = \frac{\exp\left(\psi_{k,\widehat{a}_{k,\mathrm{manip}}}(f(x),f(x'))\right)}{\sum_j \exp\left(\psi_{k,j}(f(x),f(x'))\right)} \ , \tag{4}$$

where we have introduced the functions $\psi_{k,j}$ to express the model's belief that $a_{k,\mathrm{manip}}$ assumes a value in discrete bin j, see also Figure 1B.

Objective function of S-TEC. We will now assume that the EC is a perfect copy of a, hence $p_{\mathrm{EC}}(a|x,x')$ assigns all probability to the true action a that was applied. Doing so, and putting the above pieces together, we obtain the objective that we use for experiments:

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

where we introduced x'' to always represent an input that is related to x through $a_{\mathrm{id}} = 0$. As a result, the above becomes an upper bound to equation (2), but allows us to conveniently use the same

Table 1: Accuracies obtained with linear classification with SimCLR, SimSiam & AugSelf. † denotes training for 200 epochs instead of 1,000.

Architecture	Method	CIFAR-10	CIFAR-100	STL-10
ResNet-18	SimSiam (Chen and He, 2021)	-	-	86.03†
	SimSiam + AugSelf (Lee et al., 2021)	-	-	85.98†
	SimCLR (Chen et al., 2020a)	91.48	65.33	91.61
	S-TEC (ours)	**92.04**	**66.63**	**91.94**
ResNet-50	SimCLR (Chen et al., 2020a)	93.24	70.65	94.02
	S-TEC (ours)	**93.94**	**71.88**	**94.42**

Table 2: ImageNet results (ResNet-50).

Method	Top-1 (val.)
SimCLR (Chen et al., 2020a)	64.60
S-TEC (ours)	**64.80**

Table 3: Comparison with ReLIC (ResNet-50).

Method	CIFAR-100
ReLIC (Mitrovic et al., 2021)	71.93
S-TEC* (ours)	**72.68**

method of Chen et al. (2020a) for learning the identity-related inverse model, see Appendix D for further clarification. We use x'' also for the second term, relating to 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 $a_{k,\mathrm{manip}}$, relating x and x'', falls.

4 Experimental evaluation

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 learning with our approach S-TEC. For this purpose, we considered various image datasets, including CIFAR-10/100 (Krizhevsky, 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) and ReLIC (Mitrovic et al., 2021). We follow the same procedure for all experiments, where we first performed SSL with S-TEC, or a different method for comparison, and subsequently determine the class prediction accuracy that a linear classifier can achieve. This linear classifier is trained using the representations that emerged by the preceding SSL procedure. 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 Figure 1B, and Appendix B for architectural details. These networks were optimized during SSL by gradient-descent with 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. We refer to Appendix C for details to the optimization with SSL and for details of training the linear classifier.

CIFAR-10/100 We initiated our experiments with the datasets of CIFAR-10 and 100. The accuracies that linear classifiers could attain are reported in Table 1. Our findings reveal that S-TEC increases the accuracy of a linear classifier in all settings we considered, although we conjectured originally that the images in these datasets are too small-sized, with a resolution of 32 by 32, to enable the manipulation-related inverse model to learn its task well.

We highlight that S-TEC can also be instantiated with more recent methods for SSL, such as ReLIC (Mitrovic et al., 2021), and we performed corresponding experiments on CIFAR-100 with the resulting accuracies of linear classifiers reported in Table 3. To distinguish the different methodology (i.e. target networks etc.), we denoted the specific version of our approach with S-TEC*, see also Appendix B.

STL-10 Subsequently, we studied the efficacy of S-TEC on the basis of STL-10, a dataset directly conceived and intended for developing SSL methods. Its images have an enlarged size of 96 by 96 pixels. We provide evidence in Table 1 that S-TEC improves the representations for linear classifiers in this more scaled-up setting as well.

For completeness, we added the best performing results of the approach by Lee et al. (2021), noting that we trained our networks much longer (i.e. 1,000 epochs as opposed to 200). However, we observed that their method led to improvements for STL-10 only if stronger augmentations, like solarization (which we do use, see Appendix A) were absent, see also Table 7 in their work.

ImageNet Finally, we performed experiments on ImageNet ILSVRC-2012 (Russakovsky et al., 2015) to demonstrate that S-TEC also scales gracefully to larger datasets. Here, SSL was performed for 100 epochs and report the validation accuracies that a linear classifier could achieve in Table 2 both for S-TEC and SimCLR. The result confirms that S-TEC is not restricted to small datasets.

Overall we found that our approach consistently increased the quality of features, as measured by linear classification, across a variety of datasets and network architectures. We report error bars in Appendix.

5 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 stands a family of 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.

Bootstrapping SSL. Meanwhile, several other studies proposed to train the model to produce invariant and consistent representations under content-preserving input transformations (Grill et al., 2020; Chen and He, 2021), not relying on contrasts to negative examples.

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 1.

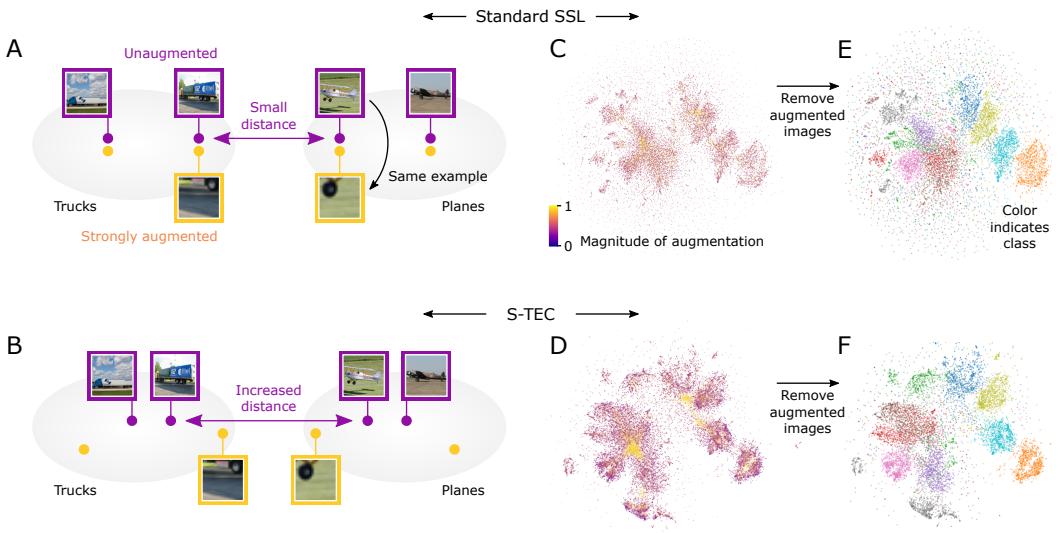

Figure 3: Analysis. A-B) Schematic of conjectured feature space organization. C-D) UMAP embeddings of images and their augmentations. E-F) UMAP embeddings of unaugmented images, using the projection of C-D.

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 2. Moreover, we also confirmed that the benefits of our approach transfer to other, orthogonal improvements, such as provided by ReLIC, see Table 3.

6 Analysis and intuitions

Our findings indicate that the objective for S-TEC, see equation (5), increases the classification performance of linear classifiers. Here, we provide our intuitions about the reasons from several perspectives, ground these by an analysis, and refer to similar arguments in the literature.

Increased information content. An obvious 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. In particular, 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. The conclusion was that this effect is caused 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 these additional nonlinear projections was increased. In summary, this observation reinforces 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. A different perspective potentially opens up new avenues to improve SSL: We conjecture that learning the additional manipulation-related inverse model using $\mathcal{L}_{\text{manip}}$ for S-TEC, see equation (2), promotes the feature space to be better organized. Firstly, note

that contrastive SSL promotes local clusters of representations that group single images and their augmentations (Wang and Liu, 2021), see Figure 3A. In turn, the single images will be linked to each other by means of overlapping augmentations (Mitrovic et al., 2021) and form extended clusters that represent classes. The representations of these links, which are likely given by strongly augmented (cropped) images, are placed at the convenience of the SSL objective, and can potentially also settle within the center region of specific classes. We argue that it is semantically more meaningful if such link-forming, potentially strongly augmented, images are located near nascent class borders. Thus, creating more sensible bridges between these boundaries, while at the same time, **increasing the distances between clusters of unaugmented ones**, see Figure 3B.

Consider the example in Figure 3A, B, which visualizes images depicting a truck with its associated "tire" augmentation, and analogously, a plane with its "tire" augmentation. In that example, S-TEC implicitly encourages representations of more strongly augmented images, the "truck tire", to be located closer to the border, since they share a similar magnitude of augmentation with strongly augmented images of other class, the "plane tire", due to the manipulation-related inverse model. Thus, the representations of images that are not distorted will be more concentrated for S-TEC (Panel B), as compared to standard SSL that are not aware of the magnitude of augmentation (Panel A).

We provide empirical evidence that supports this hypothesis in Figure 3C, D, E and 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 color-coded the magnitude of augmentation (defined as 1 minus the relative area of the cropped image). Compare Figure 3C and D. Evidently, 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, as we would expect, while the model trained with a standard SSL objective (SimCLR in this case) is oblivious to it (Panel C). If we compute the embeddings of the original, unaugmented, images using the projection obtained beforehand, as displayed in Figure 3E and F, the apparent class center distance increases for S-TEC (Panel F), due to the now missing augmented images on the border.

Moreover, we also confirmed that the benefits of our approach transfer to other, orthogonal improvements, such as provided by ReLIC, see Table 3.

7 Conclusion

In present work, we provided a generalized view on SSL that is obtained by a concept from Neuroscience, i.e. ECs, see Section 2. Specifically, we derived from first principles, see equation (1), how contemporary SSL, such as contrastive approaches, can be recovered by inverse models for ECs. Moreover, the EC reflects a richer feedback than a binary related or unrelated signal that is typically employed in current SSL. In our approach, S-TEC, this manifests itself through the training of a manipulation-related inverse model that is augmentation-aware, see Section 3. We showed experimentally in Section 4 that this insight improves the performance of recent strong baselines for SSL. Additionally, we provided our intuitions as to why S-TEC improves performance on downstream classification tasks, and grounded these empirically in Section 6. In summary, we believe that S-TEC provides a conceptually different angle from which SSL can be further improved.

Limitations. We have shown that S-TEC generalizes and extends contrastive SSL as proposed by Chen et al. (2020a); Mitrovic et al. (2021). In the future, we would like to address this limitation, and show similar improvements based on ReLICv2 (Tomasev et al., 2022). In addition, we have employed a limited set of actions (given by basic augmentations used for SSL). We believe further improvements can be obtained by introduction of broader actions/augmentations. Another potential limitation is that we have explored the improvements solely for ResNet architectures (He et al., 2016). Hence, a target for future research is to incorporate architectural improvements, and demonstrate the advantage of S-TEC in the presence of self-attention (Vaswani et al., 2017).

Potential negative societal impacts. Our approach S-TEC provides an improvement to contemporary SSL, thus inheriting potential negative societal impacts of SSL. Since S-TEC, as also SSL, can be applied to unlabelled (uncurated) data, our concern is that a malicious application of S-TEC to health databases can help adversaries to exploit patients, by creating better profiles through the richer feedback that we propose.

References

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.

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.

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.

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.

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. (2009). Learning multiple layers of features from tiny images. Technical report.

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.

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*.

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*.

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]