SIMPLICIAL EMBEDDINGS IN SELF-SUPERVISED LEARNING AND DOWNSTREAM CLASSIFICATION

Samuel Lavoie[◇*], **Christos Tsirigotis**[◇], **Max Schwarzer**[◇], **Ankit Vani**[◇], **Michael Noukhovitch**[◇], **Kenji Kawaguchi**[‡], **Aaron Courville**[◇♣]
[◇] Mila, Université de Montréal, [‡] National University of Singapore, [♣] CIFAR Fellow

ABSTRACT

Simplicial Embeddings (SEM) are representations learned through self-supervised learning (SSL), wherein a representation is projected into L simplices of V dimensions each using a softmax operation. This procedure conditions the representation onto a constrained space during pre-training and imparts an inductive bias for discrete representations. For downstream classification, we provide an upper bound and argue that using SEM leads to a better expected error than the unnormalized representation. Furthermore, we empirically demonstrate that SSL methods trained with SEMs have improved generalization on natural image datasets such as CIFAR-100 and ImageNet. Finally, when used in a downstream classification task, we show that SEM features exhibit emergent semantic coherence where small groups of learned features are distinctly predictive of semantically-relevant classes.

1 INTRODUCTION

Self-supervised learning (SSL) is an emerging family of methods that aim to learn representations of data without manual supervision, such as class labels. Recent works (Hjelm et al., 2019; Grill et al., 2020; Saeed et al., 2020; You et al., 2020) learn dense representations that can solve complex tasks by simply fitting a linear model on top of the learned representation. While SSL is already highly effective, we show that changing the *type* of representation learned can improve both the performance and interpretability of these methods.

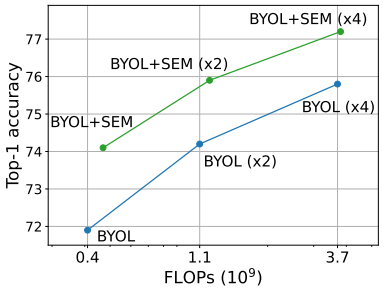


Figure 1: Linear probe accuracy of BYOL and BYOL + SEM on ImageNet trained for *200 epochs* with a ResNet-50 architecture.

For this we draw inspiration from overcomplete representations: representations of an input that are non-unique combinations of a number of basis vectors greater than the input's dimensionality (Lewicki & Sejnowski, 2000). Mostly studied in the context of the sparse coding literature (Gregor & LeCun, 2010; Goodfellow et al., 2012; Olshausen, 2013), sparse overcomplete representations have been shown to increase stability in the presence of noise (Donoho et al., 2006), have applications in neuroscience (Olshausen & Field, 1996; Lee et al., 2007), and lead to more interpretable representations (Murphy et al., 2012; Fyshe et al., 2015; Faruqui et al., 2015). But, the basis vector is learned using linear models (Lewicki & Sejnowski, 2000; Teh et al., 2003).

In this work, we show that SSL may be used to learn discrete, sparse and overcomplete representations. Prior work has considered sparse representation but not sparse and overcomplete representation learning with SSL; for example, Dessì et al. (2021) propose to discretize the output of the encoder in a SSL model using Gumbel-Softmax (Jang et al., 2017). However, we show that discretization during pre-training is not necessary to achieve a sparse representation. Instead, we propose to project the encoder's output into L vectors of V dimensions onto which we apply a softmax function to impart an inductive bias toward sparse one-hot vectors (Correia et al., 2019; Goyal et al., 2022), also alleviating the need to use high-variance gradient estimators to train the encoder. We refer to this embedding as Simplicial Embeddings (SEM), as the softmax functions map the unnormalized representations onto L simplices. The procedure to induce SEM is simple, efficient, and generally applicable.

[*]Correspondence to: samuel.lavoie.m@gmail.com

The SSL pre-training phase, used with SEM, learns a set of L *approximately* one-hot vectors. Key to controlling the inductive bias of SEM during pre-training is the softmax temperature parameter: the lower the temperature, the stronger the bias toward sparsity. Consistent with earlier attempts at sparse representation learning (Coates & Ng, 2011), we find that the optimal sparsity for pre-training need not match the optimal level for downstream learning.

For downstream classification, we may discretize the learned representation by, for example, taking the argmax for each simplex. But, we can also use SEM to control the representation's expressivity via the softmax's temperature. We provide a theoretical bound showing that the expected error follows a trade-off between the training error and the representations' expressivity that can be controlled by the softmax's temperature used to normalize the representation for downstream classification. Our bound also shows improved expected error as we increase L and V for SEM.

SEM is generally applicable to recent SSL methods. Applying it to seven different SSL methods (Chen et al., 2020b; He et al., 2020; Grill et al., 2020; Caron et al., 2020; 2021; Zbontar et al., 2021; Bardes et al., 2022), we find accuracy increases of 2% to 4% on CIFAR-100. We observe monotonic improvement as we increase the number of vectors L, showing the benefit of the overcomplete representations learned by SEM, while this improvement is absent when we do not use softmax normalization. When training a SSL method with SEM on ImageNet we also observe improvements on in-distribution compared to the baseline (Figure 1). We also observe improvement on out-of-distribution test sets, semi-supervised learning benchmark and transfer learning datasets, demonstrating the potential of SEM for large scale applications. Finally, we find that SEM learns features that are closely aligned to the semantic categories in the data. This demonstrates that SEM learns disentangled and interpretable representations, as previously observed in overcomplete representations (Faruqui et al., 2015).

2 Related work

The softmax operation has been used in other contexts, notably as an architectural component for models to attend to context-dependent queries via, for example, an attention mechanism (Bahdanau et al., 2016; Vaswani et al., 2017; Correia et al., 2019; Goyal et al., 2022), a mixture of experts (Jordan & Jacobs, 1993) or memory augmented networks (Graves et al., 2014). This operation is also used for the computation of several SSL objectives such as InfoNCE (van den Oord et al., 2018; Hjelm et al., 2019), and as a normalization of the output to compute the objective in DINO and SWaV (Caron et al., 2020; 2021). Different from these, our method places the softmax at the output of an encoder to constrain the representation into a set of L sparse vectors.

Similar to our approach, other architectural constraints such as Dropout (Srivastava et al., 2014), BatchNorm (Ioffe & Szegedy, 2015) and LayerNorm (Ba et al., 2016) also improve the training of large neural networks. However, contrary to SEMs, they are not used to induce sparsity on the representation or control its expressivity for downstream tasks. Closer to our work, Liu et al. (2021) propose to constrain the expressivity of the representation of a neural network with a set of discrete-valued symbols obtained using a set of Vector Quantized (Oord et al., 2018) bottlenecks. Similarly, Dessi et al. (2021) propose a communication game with a discrete bottleneck. The idea of discretizing the encoder's output is similar to using SEM vectors that are one-hot (e.g. temperature $= 0$) and only one symbol (e.g. $L = 1, V = 2048$). In our work, we find success in removing the hard-discretization and having $L > 1$, which can be interepreted as combining several symbols.

3 Simplicial Embeddings

Simplicial Embeddings (SEM) are representations that can be integrated easily into a contrastive learning model (Hjelm et al., 2019; Chen et al., 2020b), the BYOL method (Grill et al., 2020), and other SSL methods (Caron et al., 2020; 2021; Zbontar et al., 2021). For example, in BYOL, we insert the SEM after the encoder and before the projector and the rest is unchanged as shown in Figure 2c. In this figure, t and t' are augmentations defined by the practitioner, ξ are parameters of the target network that are updated as moving average of the parameters θ of the online networks trained with SGD. So, ξ are updated as follow: $\xi \quad \alpha\xi + (1 \quad \alpha)\theta$, with $\alpha \; 2 \; [0, 1]$.

To produce SEM representation, the encoder's output e is embedded into L vectors $z_i \; 2 \; \mathsf{R}^V$. A temperature parameter τ scales z_i, and then a softmax re-normalizes each vector z_i to produce z_i.

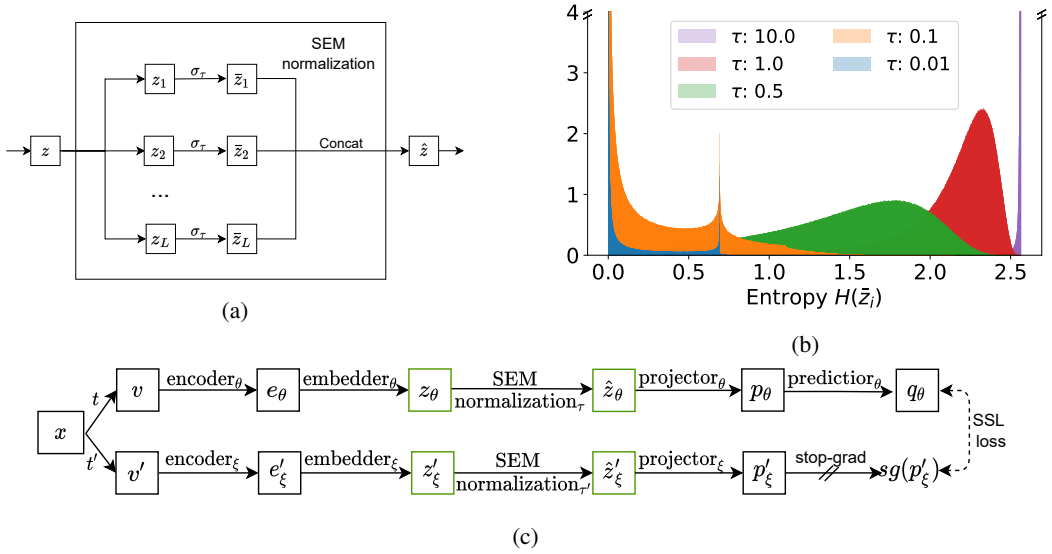


Figure 2: **(a)** Procedure to obtain Simplicial Embeddings (SEM). A matrix $z \in \mathbb{R}^{L \times V}$ contains L vectors $z_i \in \mathbb{R}^V$. The vectors z_i are normalized with σ , the softmax operation with temperature τ. The normalized vectors are concatenated into the vector $\hat{z}$. **(b)** Normalized histogram of the entropies $H(z_i)$ of each simplex z_i for the sample in CIFAR's training dataset at the end of pre-training with various τ. The peak at $\ln(2)$ for $\tau = 0.01$ and $\tau = 0.1$ are a large number of simplices with two elements close to 0.5. **(c)** Integration of SEM with BYOL (Grill et al., 2020). The encoder outputs a latent vector which is embedded into the matrix $z \in \mathbb{R}^{L \times V}$ and then transformed into SEM.

Finally, the normalized vectors z_i are concatenated to produce the vector $\hat{z}$ of length $L \cdot V$. We illustrate SEM in Figure 2a. Formally, the re-normalization is as follows:

$$z_i := \sigma(z_i), \quad \sigma(z_i)_j = \frac{e^{z_{ij}/\tau}}{\sum_{k=1}^{V} e^{z_{ik}/\tau}}, \quad \hat{z} := \text{Concat}(z_1, \ldots, z_L), \quad \forall i \in [L], \forall j \in [V]. \tag{1}$$

3.1 Inductive bias towards sparsity during pre-training

In SEM, L controls the numbers of simplices and V controls the dimensionality of each simplex. As such, the higher V is, the sparser the representation can be. During pre-training, the constraint induced by embedding the representation into a simplex biases each vector towards sparse vectors by creating a zero-sum competition between the components of the vector. In order for a component to increase by α, then the other elements must decrease by α, and all elements are bounded by 0. For networks to learn useful features and minimize their objective, they must prioritize some components at the expense of others. The strength of this bias is controlled via the pretraining temperature τ_p of the softmax, and the size of the vectors V as it was noted in the context of attention (Vaswani et al., 2017; Wang et al., 2021b). For SSL methods with a target network, the temperature for the target network can be different to the online network's as no gradient is back-propagated through it.

To visualize the effect of the temperature on SEM after pre-training, we interpret each simplex as a probability mass function $p(z_{ij})$ where, for all $i \in [L]$, $\sum_{j=1}^{V} p(z_{ij}) = 1$ and $p(z_{ij}) \geq 0 \; \forall j$. The entropy of a simplex z_i, defined as $H(z_i) := \sum_{j=1}^{V} p(z_{ij}) \log p(z_{ij})$, informs whether the simplex is a sparse or a dense vector. That is, if $H(z_i^{(x)}) = 0$ then the vector is one-hot. On the other hand, if $H(z_i^{(x)}) = \ln(V)$ then the vector is dense and uniform. While the temperature τ_p is merely a scaling of the logits, it has an important control over the learned representation's entropy and resulting SEM sparsity. We demonstrate this by learning a representation on CIFAR-100 using BYOL, and analyze the entropies of the resulting simplices. In Figure 2b, we plot the histogram of the entropies $H(z_i)$, for a given τ_p, of each simplex for each sample in the training set of CIFAR-100. We observe that

even after pre-training, small temperatures ($\tau_p = 0.01$) yields representations that are close to one-hot vectors while high temperatures yields vectors that are close to uniform vectors.

By pre-training using a softmax, SEMs create representations that are conditioned to fit onto simplices. In pre-training, we select τ_p for optimal inductive bias: τ_p too small yields vanishing gradients (Wang et al., 2021b) and τ_p too large yields a bias that is too weak. We may select a different optimal τ_d for downstream performance as discussed formally in the next subsection.

3.2 SEM IMPROVEMENT ON THE GENERALIZATION OF THE DOWNSTREAM CLASSIFIER

In this subsection, we theoretically demonstrate the benefit of training a downstream classifier with SEM normalized input compared to a baseline classifier with unnormalized input. We show that: (1) there is a trade-off between the training loss and the generalization gap, which is controlled by the value of τ_d (denoted $\tau := \tau_d$ in this subsection), (2) SEM can improve the base model performance when we attain good balance in this trade-off, and (3) the improvement due to SEM is expected to increase or stay constant as L and V increase. In the remainder of this subsection, we introduce the notation and assumptions needed to understand and derive the result, then present our theoretical claim and discuss its implications.

Notation. We use a training dataset $S = (z^{(i)}, y^{(i)})_{i=1}^{n}$ of n samples for supervised training of a classifier, using the representation z extracted from the pre-trained model* and the corresponding label $y \in Y$ where Y is the space of possible labels. Assume that $z \in Z = [-1, +1]^{L \times V}$, which means that z is a matrix with L rows and V columns. We denote the element of z at row i and column j as z_{ij}. Let g represent the downstream classifier. We refer to the baseline downstream model with unnormalized input as f_{base}, and $f_{\text{base}}(z) = g(z)$. The corresponding downstream model trained with the SEM normalization is $f_{\text{SEM}(\tau)}(z) = (g \circ \sigma_\tau)(z)$, where σ_τ is applied element-wise along each row of z such that $\sigma_\tau(z_{ij}) = \frac{e^{z_{ij}/\tau}}{\sum_{t=1}^{V} e^{z_{it}/\tau}}$ for $j = 1, \ldots, V$. Moreover, we define f^S_{base} and $f^S_{\text{SEM}(\tau)}$ the base and the SEM normalized models obtained by fitting the dataset S. Finally, let H be the union of the hypothesis spaces of $f_{\text{SEM}(\tau)}$ and f_{base}.

To compare the quality of the base model and the model with SEM normalization, we analyze the generalization gap $\mathbb{E}_{z,y}[l(f_S(z), y)] - \frac{1}{n}\sum_{i=1}^{n} l(f_S(z^{(i)}), y^{(i)})$ for each $f_S \in \{f^S_{\text{SEM}(\tau)}, f^S_{\text{base}}\}$, where $l : \mathbb{R} \times Y \to \mathbb{R}_{\geq 0}$ is the per-sample loss.

The key insight that we exploit for the theorem is that the softmax operation σ_τ controls the expressivity of the input's representation to g via the temperature τ. We denote $\varphi_{f_{\text{base}}}$ as an upper bound on the expressivity of z_i for the baseline model f_{base}, and $\varphi_{f_{\text{SEM}(\tau)}}$ as the upper bound on the expressivity of $\sigma_\tau(z_i)$ for the model with SEM normalization $f_{\text{SEM}(\tau)}$. The formal definition of $\varphi_{f_{\text{base}}}$ and $\varphi_{f_{\text{SEM}(\tau)}}$ requires proof devices that will hinder the readability of this section, so we refer the reader to Appendix A for a detailed definition. Let $\varphi_f \in \{\varphi_{f_{\text{base}}}, \varphi_{f_{\text{SEM}(\tau)}}\}$. Intuitively, φ_{f_S} measures the largest possible distance that two embeddings can have such that the largest component remains the same for both embeddings. We note that this measure depends only on V for f_{base}, and on both V and τ for $f_{\text{SEM}(\tau)}$. We use $\varphi_{f_S}(V, \tau)$ to denote the measure given by either model and note that τ has no effect for f_{base}.

Assumptions. We assume that the per-sample loss is bounded such that $l(f(z), y) \leq B$ for all $f \in H$ and for all $(z, y) \in Z \times Y$. For example, $B = 1$ for the 0-1 loss. Next, let l_y be the per-sample loss given y. We assume that $l_y \circ g$ are uniformly Lipschitz functions for all $y \in Y$ and $g \in G_S$, where G_S is the set of classifiers g returned by the training algorithm using the dataset S. Let R be such a uniform Lipschitz constant. This means that $|(l_y \circ g)(\sigma_f(z)) - (l_y \circ g)(\sigma_f(z'))| \leq R\|\sigma_f(z) - \sigma_f(z')\|_F$, where $l_y(g \circ \sigma_f(z)) = l(g \circ \sigma_f(z), y)$, and $\sigma_f = \sigma_\tau$ when $f = f_{\text{SEM}(\tau)}$ and σ_f is identity when $f = f_{\text{base}}$. Finally, we assume that there exists $\varepsilon > 0$ such that for all representations z of the underlying distribution we have that for any $i \in [L]$, if $k = \arg\max_{j \in [V]} z_{ij}$, then $z_{ik} \geq z_{ij} + \varepsilon$ for any $j \neq k$. Since ε can be arbitrarily small (e.g. as small as machine precision), this assumption typically holds in practice. We are now ready to state our theoretical claim.

Theorem 1 illuminates the advantage of SEM and the effect of the hyper-parameter τ on the performance of the downstream classifier. We present the proof in Appendix A.

*In this subsection, we refer to the extracted representation as z, the embedder's output

Theorem 1. *Let $V \geq 2$. For any $1 > \delta > 0$, with probability at least $1 - \delta$, the following holds for any $f_S \in \{f^S_{\text{SEM}(\cdot)}, f^S_{\text{base}}\}$:*

$$\mathbb{E}_{z,y}[l(f_S(z), y)] \leq \frac{1}{n}\sum_{i=1}^{n} l(f_S(z^{(i)}), y^{(i)}) + R\sqrt{L\,\varphi_{f_S}(V,\tau)} + c\sqrt{\frac{\ln(2/\delta)}{n}},$$

where $c > 0$ is a constant in $(n, f, H, \delta, \tau, S)$. Moreover,

$$\varphi_{f^S_{\text{SEM}(\cdot)}} \to 0 \text{ as } \tau \to 0 \quad \text{and} \quad \varphi_{f^S_{\text{SEM}(\cdot)}} - \varphi_{f^S_{\text{base}}} \leq \frac{3}{4}(1 - V) < 0 \quad \forall \tau > 0.$$

The first statement of Theorem 1 shows that the expected loss is bounded by the three terms: the training loss $\frac{1}{n}\sum_{i=1}^{n} l(f_S(z^{(i)}), y^{(i)})$, the second term $R\sqrt{L\varphi_{f_S}}$, and the third term $c\sqrt{\frac{\ln(2/\delta)}{n}}$. Since c is a constant in $(n, f, H, \delta, \tau, S)$, the third term goes to zero as $n \to \infty$ and is the same with and without SEM. Thus, for the purpose of assessing the impact of SEM, we can focus on the second term, where a difference arises. Theorem 1 shows that $R\sqrt{L\varphi_{f_S}}$ goes to zero with SEM; i.e., $\varphi(f^S_{\text{SEM}(\cdot)}) \to 0$ as $\tau \to 0$. Also, for any $\tau > 0$, the second term with SEM is strictly smaller than that without SEM as $\varphi_{f^S_{\text{SEM}(\cdot)}} - \varphi_{f^S_{\text{base}}} \leq \frac{3}{4}(1 - V) < 0$ and demonstrates that the improvement due to SEM is expected to asymptotically increase as V increases. Moreover, L is a multiplicative constant of φ which shows that, as L increases, the improvement due to SEM is also expected to be higher. Overall, Theorem 1 shows the benefit of SEM as well as the trade-off with τ. When $\tau \to 0$, the second term goes to zero, but the training loss (the first term) can increase due to underfitting resulting from the reduction in representation expressivity. Thus, τ should be chosen to optimally balance this trade-off.

4 Empirical analysis

We empirically study the effect of SEM on the representation of SSL methods and demonstrate that SEM improves the test set accuracy on CIFAR-100 (Krizhevsky, 2009). We compare SEM with other methods for inducing sparse representations during pretraining and demonstrate that SEM lead to better downstream accuracy. On ImageNet (Deng et al., 2009), we study the effect of SEM on robustness, semi-supervised learning and transfer learning datasets, demonstrating consistent improvement attributed to SEM. Finally, we present evidences that features produced by SEMs are more naturally aligned with the semantic categories of the data. The code for reproducing the results is available at: https://github.com/lavoiems/simplicial-embeddings/.

Training setup. For all experiments, we build off the implementation of the baseline models from the Solo-Learn library (da Costa et al., 2021). We probe the encoder's output for the baseline methods, as typically done in the literature. For models with SEM, we probe the SEM normalized representation (i.e. $\hat{z}$). In our experiments, the embedder is a linear layer followed by BatchNorm (Ioffe & Szegedy, 2015). Unless mentioned otherwise, we use $L = 5000$ and $V = 13$ for the SEM representation. We do not perform any search for the non-SEM hyper-parameters. The SEM hyper-parameters are selected by using a validation set of 10% of the training set of CIFAR-100 and 10 samples per class on the in distribution dataset for ImageNet. The test accuracy is obtained by retraining the model with all of the training data using the parameters found with the validation set. We pre-train the SSL models for 200 epochs on ImageNet and 1000 epochs on CIFAR-100.

Table 1: Linear probe top-1 accuracy on CIFAR-100 trained for *1000 epochs* with a ResNet-18/50 encoder. We compare the *test accuracy* of several SSL models with and without SEM. **Boldface** indicates highest accuracy. Green rows indicate a SSL method + SEM.

	SimCLR	MoCo	BYOL	Barlow-Twins	SwAV	DINO	VicReg
ResNet-18:							
Baseline	66.8	69.3	70.7	70.7	64.6	66.8	68.5
With SEM	**69.5**	**71.0**	**73.9**	**73.0**	**67.7**	**69.2**	**71.4**
ResNet-50:							
Baseline	70.5	73.24	74.2	72.0	–	–	70.8
With SEM	**73.2**	**75.8**	**77.5**	**73.3**	–	–	**73.3**

4.1 SEM IMPROVES ON DOWNSTREAM CLASSIFICATION

Baseline comparison. We evaluate the effect of adding SEMs in seven modern SSL approaches. We take standard SimCLR (Chen et al., 2020b), MoCo-v2 (He et al., 2020), BYOL (Grill et al., 2020) Barlow-Twins (Zbontar et al., 2021), SwAV (Caron et al., 2020), DINO (Caron et al., 2021) and VicReg (Bardes et al., 2022) models and implement SEM after the encoder. We compare our approach on CIFAR-100 with a ResNet-18 and ResNet-50 in Table 1. We found SWaV and DINO to be unstable with ResNet-50 thus have decided not to compare them with SEM. For every SSL methods, using SEMs improves the baseline methods by 2% to 4% demonstrating that SEM is a general approach that improves in-distribution generalization for SSL methods.

Increasing the representation's size of SEM increases the performance. We find that increasing L (the number of simplices of SEM) beyond the over-complete regime increases the downstream accuracy. This increased performance is not observed when we abstain from using the softmax normalization of SEM. In Figure 3, using a ResNet-50 encoder, we compare BYOL + SEM, with an identical model without the Softmax normalization which we call BYOL + Embed and BYOL to which we increase the representation's size before the mean-pooling using the method proposed in (Dubois et al., 2022) and described in their Appendix F. To be clear, the extracted representation of BYOL + Embed is the embedder's output z and the extracted representation for BYOL is the encoder's output e. We use $V = 13$ and scale $L \in [10, 10000]$ to get a range of representation sizes.

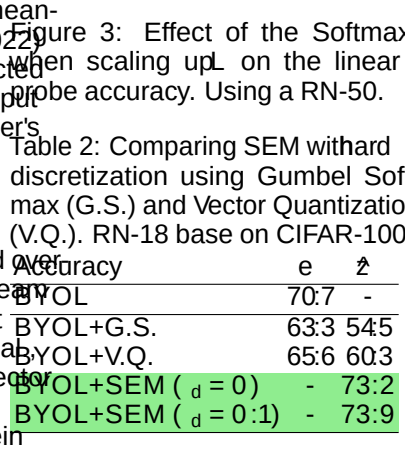


Figure 3: Effect of the Softmax when scaling up L on the linear probe accuracy. Using a RN-50.

Comparison of SEM with hard discretization approaches. Several other methods can be used to induce a sparse and over-complete representation during pre-training and downstream classification. For example, we may sample discrete one-hot codes of V dimensions using Gumbel Softmax (Jang et al., 2017) as done in Dessì et al. (2021). We can also use Vector Quantization (VQ) (Oord et al., 2018) and consider L latent embedding spaces with V embedding vectors each, wherein the vectors are in $\mathbb{R}^d$. In contrast to SEM, it is not possible to propagate the gradient through the bottleneck trivially and VQ uses straight-through estimation in the embedding space to back-propagate the gradient to the encoder. Here, we observe that these alternative approaches exhibit a considerable decrease in performance in comparison to the baseline as demonstrated in Table 2. In this table, we reproduce the same setup as SEM but we replace the Softmax with hard discretization baselines methods. For discretization with Gumbel Straight-Through estimation, we use the same setup as SEM with $L = 5000$ and $V = 13$, that is 5000 one-hot vectors of 13 dimensions and $= 2$[†]. For VQ, we found that $L = 512$ and $V = 128$ led to the best performance. That is, we have 512 latent embedding spaces, each with 128 possible embedding vectors that are in $\mathbb{R}^{32}$.

Table 2: Comparing SEM with hard discretization using Gumbel Softmax (G.S.) and Vector Quantization (V.Q.). RN-18 base on CIFAR-100.

Accuracy	e	z
BYOL	70.7	-
BYOL+G.S.	63.3	54.5
BYOL+V.Q.	65.6	60.3
BYOL+SEM ($_d = 0$)	-	73.2
BYOL+SEM ($_d = 0.1$)	-	73.9

We note that while we have not found hard-discretization to be successful during pre-training, we may hard-discretize a SEM representation for downstream task. In Table 2, we also present SEM with $_{DS} = 0$, which correspond to using the discretized representation for downstream classification. We obtain the discrete representation by taking the argmax for each simplex. This result demonstrating that SEM with pre-training can be used to learn meaningful discrete codes for downstream applications and yields better performance than the baselines, implying that pre-training with SEM could be be used in applications that require discretization.

Memory and computational efficiency of SEM. SEM's performance improvements come at a cost of increased memory allocation (VRAM) due to additional parameters needed to perform the matrix multiplication, and slightly more computation (FLOPs/sample). For very large over-complete representation the increased memory requirement can impede practical application. We propose a more efficient version of SEM by sparsifying the matrix multiplication of the embedder and of the projector and detail this procedure in Appendix D.1. As shown in Table 17, SEM with sparse matrix

[†] A hyper-parameter search was performed to select the best performing hyper-parameter.

multiplication use only slightly more memory and compute but outperforms the BYOL baseline on CIFAR-100 though underperforming the regular SEM. We also note that SEM's memory cost becomes relatively minor as we scale up the encoder. As well, the computational cost of SEM is small compared to the total cost of pre-training and achieves higher accuracy using fewer FLOPs compared to scaling the encoder as shown in Figure 1.

4.2 Analyzing the parameters of SEM

We present two figures in this section to better understand the effect of the parameters of SEM on the downstream accuracy. In Figure 4, we evaluate the effect of changing τ_p and τ_d on the downstream accuracy. In Figure 5, we evaluate the effect of L and V on the downstream accuracy and also contrast f_{base} and $f_{SEM}(\cdot\,; \lambda = 1)$ by using the same encoder pre-trained with SEM. This allows us to relate some observations to the theorey presented in Section 3.2. Now, we discuss the effect of each of SEM's parameter on the resulting downstream classification.

Increasing V yields a steep performance increase for small V but quickly plateau. In Figure 5b, we observe a steep increase of the accuracy for $V < 13$ followed by a plateau for $V > 13$. In Figure 4a, we observe that the optimal accuracy obtained for $V = 1024$ and $L = 64$ is similar to the one obtained for $L = 50$ (Embedding size=650) in Figure 3.

Increasing L yields monotonical improvement for downstream classification. In the regime that we can test it, increasing L lead to consistent improvement on the downstream accuracy as observed in Figure 3 and Figure 5a. Using SEM in pre-training only is not enough and using it in the downstream classifier is necessary for the improved performance as demonstrated in Figure 5a.

The optimal τ_p depends on V. As previously noted in the context of Attention (Vaswani et al., 2017; Wang et al., 2021a), the optimal attention's temperature is proportional to attention's vector size. We also observe this in SEM. As presented in Figure 4a, the optimal τ_p for larger V is higher.

Models with larger L are more robust to smaller τ_d. In Figure 4, we observe that SSL models are more robust to smaller τ_d as L increase. We speculate that the information can be scattered across the simplices for large L, allowing to reduce the expressivity of each vector with minimal impact on the downstream accuracy.

(a) (b)

Figure 4: Effect of τ_p and τ_d on a RN-50.

(a) (b)

Figure 5: Comparing f_{SEM} and f_{base} on a RN-18.

4.3 SEM improvement on large-scale datasets with ImageNet

Figure 1 in the introduction demonstrates that using SEM leads to better in distribution generalization for ImageNet and is a more efficient method of scaling up the model as compared to scaling up the width of the ResNet-50 encoder. Here, we demonstrate that SEM generally improves the accuracy on several robustness test sets, a semi-supervised learning benchmark and transfer learning datasets. We use BYOL+SEM with an embedding size of 105 000 features ($L = 5000$ and $V = 21$) for these experiments. The embedding is pre-trained for 200 epochs using the BYOL SSL procedure.

Robustness to out-of-distribution test sets We perform a comparative study using several test sets: (IN) the in-distribution test set provided in ImageNet; (IN-C) ImageNet-C, which exhibits a set of common image corruptions (Hendrycks & Dietterich, 2019); (IN-R) ImageNet-R (Hendrycks et al., 2021) which consists of different renderings for several ImageNet classes; and (IN-V2) ImageNet-V2 (Recht et al., 2019), a distinct test set for ImageNet collected using the same process as IN. (IN-A) ImageNet-A (Chen et al., 2020a) contains a set of samples that are miclassified by a ResNet-50 classifier. We use the methodology and software proposed in Djolonga et al. (2020; 2021)

Table 3: Robustness via linear probe top-1 test accuracies on IMAGENET variant datasets, using representations pre-trained for 200 epochs. * Taken from (Chen & He, 2020)

	IN	IN-V2	IN-R	IN-C	IN-A
BYOL*	70.6	-	-	-	-
BYOL	71.9	59.2	18.8	39.5	1.65
BYOL+SEM	74.1	61.2	22.1	43.4	2.53

Table 4: Top-1 transfer learning accuracy from IMAGENET pre-trained representation.

	FOOD101	C10	C100	SUN	DTD	FLOWER
Linear probe:						
BYOL	74.2	91.8	74.9	60.9	72.2	88.9
BYOL+SEM	74.7	93.5	78.6	62.1	71.9	91.5
Fine-tuned:						
BYOL	83.1	97.2	83.6	59.1	69.2	85.4
BYOL+SEM	84.7	97.2	85.6	63.3	71.3	91.7

to perform our experiments. We observe that BYOL + SEM outperforms BYOL on every robustness datasets probed, demonstrating that SEM also improves generalization to out-of-distribution test sets.

Transfer learning. We probe the effectiveness of SEM in BYOL and MoCo when transferring representations trained on IMAGENET to other classification tasks. We follow the linear evaluation and fine-tuning methodologies described in previous works (Grill et al., 2020; Lee et al., 2021), which entails training a linear classifier with logistic regression using sklearn (Pedregosa et al., 2011) on the embeddings of the samples and fine-tuning the encoder respectively. To avoid out-of-memory issues that may occur in the linear probe experiment with the sklearn solver when the number of features, we discretize our features and use sparse matrix to fit the logistic regression. This is equivalent to forcing $_d = 0$ for all the experiments. For the fine-tuning experiments, we $_dx = 1$ since the evaluation method allows for mini-batch gradient descent. We perform our transfer learning experiments on the following datasets: FOOD (Bossard et al., 2014), CIFAR-10 (C-10) (Krizhevsky, 2009), CIFAR-100 (C-100) (Krizhevsky, 2009), SUN (Xiao et al., 2010), DTD (Cimpoi et al., 2014) and FLOWER (Nilsback & Zisserman, 2008).

This task evaluates the generality of the encoder as it has to encode samples from various out-of-distribution domains with categories that it may not have seen during training. We present our results in Table 4 and observe that SEM improves the transfer accuracy over the baseline for every datasets but DTD for the linear probe experiment. For DTD, we hypothesize that the drop in performance is due to the fact that we use a temperature that is too small. Since this is a texture dataset with higher frequency, it might be the case that we need more expressivity to correctly fit the data. We support the conjecture with the fine-tuning experiment where BYOL + SEM out-performs the baseline.

Semi-supervised learning. We evaluate the effect of using SEM when fine-tuning on a classification task with a small subset of IMAGENET's training set. We follow the semi-supervised learning procedure of Chen et al. (2020b); Grill et al. (2020) and use the same fixed splits of 1% and 10% of ImageNet labelled training set. In Table 5, we demonstrate that using SEM lead to an important increased performance, especially in the low supervised data regime.

Table 5: Semi-supervised learning accuracy by fine-tuning on IMAGENET.

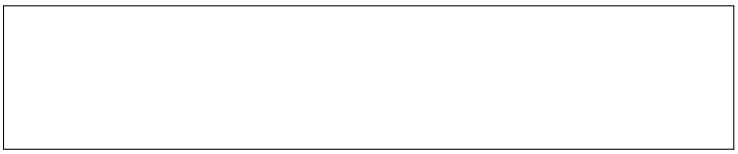
	Top-1		Top-5	
	1%	10%	1%	10%
BYOL	51.6	67.5	78.0	88.9
BYOL+SEM	56.7	69.9	81.0	90.0

4.4 SEMANTIC COHERENCE OF SEM FEATURES

Here we demonstrate that SEM features are coherently aligned with the semantics present in the training data. Qualitatively, we visualize the most predictive features of a downstream linear classifier trained on CIFAR-100 and see that the classes with similar predictive features are semantically related. Quantitatively we propose a metric that returns the ratio of features mostly predictive for a classes that are in the same super class to total number of class predictive for this feature.

For both our analysis, we use a linear classifier trained on the features extracted from BYOL with and without SEM. Consider the trained linear classifier with a weight matrix $W \in R^{N \times C}$, with N features, and C classes. By preserving the top-k parameters of the weight matrix W for each class and pruning the features predictive for only one class, we create a bipartite graph between two set of nodes: the CIFAR-100 classes and the features of the representation. We denote this graph W_{pruned}.

The qualitative analysis is given by plotting the subset W_{pruned} obtained by taking the top-5 features for each class. We present a subset of the graph for BYOL+SEM in Figure 6a and for BYOL in Figure 6b. The full graphs are presented in the Appendix. In the SEM plot, a set of connected components emerge, and the connected components of the graph are semantically related. For example, the



(a) BYOL + SEM

(b) BYOL

(c)

Figure 6: Semantic coherence of the features. (a) and (b) Subset of W_5, the bipartite graph of the most 5 highest magnitude features on BYOL + SEM features (a) and BYOL on the encoded features (b). (c) Coherence of the top K features to the semantics of the super-class of the categories of CIFAR-100. It is taken as the number of pairwise categories in the same super-class for which a feature is among its top K most predictive features over the total number of pairwise categories.

rst set of connected components are owers, and the last set of connected components are aquatic mammals[‡]. The same class coherence is not observed with either the BYOL baseline or with BYOL augmented with a large representation. In particular, we do not see a small number of semantically related connected components. Instead, we see a large fully connected graphs.

Next, we describe how we quantitatively measure the semantic coherence of the features. Notice that two classes share a common predictive feature in W_K if they are 2-neighbour. Let $N(c_i)$ returns all pairs (c_i, c_j) for all j 2-neighbour of c_i. Moreover, de ne the operation $is\_super(c_i, c_j)$ which returns 1 if c_i and c_j are from the same CIFAR-100 superclass and 0 otherwise. We reproduce the superclass of CIFAR-100 in Table 22 in the Appendix. We measure semantic coherence as follows:

$$\text{Coherence}(W_K) := \frac{1}{C} \sum_{i=1}^{C} \frac{\sum_{(c_i, c_j) \in N(c_i)} is\_super(c_i, c_j)}{|N(c_i)|}; \tag{2}$$

where $C = 100$ for CIFAR-100 and $|\cdot|$ is the cardinality of a set.

We compare the semantic coherence of BYOL+SEM with the control experiments of BYOL: regular BYOL, BYOL with an embedding of the same size as BYOL+SEM but without the normalization and BYOL to which we applied linear ICA (Hyvärinen & Oja, 2000) in an attempt to disentangle the features. In Figure 10, we plot the full graph W_5 for BYOL+SEM and the baselines. We observe that using the SEM yields semantically coherent features for all the classes of CIFAR-100. This observation is consistent with the qualitative and quantitative experiments presented earlier and demonstrates that SEM's inductive bias during pre-training leads to features that are semantically coherent with the semantic categories extant in the data. This arguably have important implications for improving the interpretability of SSL representations.

5 Conclusion

SEM is a simple, drop-in module that induces discrete sparse overcomplete representations for standard SSL methods using a softmax operation. This simple modi cation leads to improved generalization on downstream classi cation across several state-of-the-art SSL methods. Furthermore, SEM improves performance on out-of-distribution, semi-supervised, and transfer learning tasks across the board and also scales with encoder size. By analyzing semantic coherence, we nd that SEMs naturally disentangle data into semantic categories without any explicit training objectives.

[‡]Although " at sh" may seem out of place in the third set, manually checking CIFAR images showed that many images labelled " at sh" are often humans holding at sh.

ACKNOWLEDGEMENTS

The authors are grateful for the insightful discussions with Xavier Bouthillier, Hattie Zhou, Sébastien Lachapelle, Tristan Deleu, Yuchen Lu, Eeshan Dhekane, Maude Lizaire, Julien Roy and David Dobre. We acknowledge funding support from Samsung and Hitachi, as well as support from Aaron Courville's CIFAR CCAI chair. We also wish to acknowledge Mila and Compute Canada for providing the computing infrastructure that enabled this project. Finally, this project would not have been possible without the contribution of the following open source projects: Pytorch (Paszke et al., 2019), Orion (Bouthillier et al., 2022), Solo-Learn (da Costa et al., 2021), Scikit-Learn (Pedregosa et al., 2011), and Numpy (Harris et al., 2020).

REFERENCES

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. arXiv:1409.0473 [cs, stat], May 2016. URL http://arxiv.org/abs/1409.0473. arXiv: 1409.0473.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In International Conference on Learning Representations, 2022. URL https://openreview.net/forum?id=xm6YD62D1Ub.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), Computer Vision – ECCV 2014, pp. 446–461, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10599-4.

Xavier Bouthillier, Christos Tsirigotis, François Corneau-Tremblay, Thomas Schweizer, Lin Dong, Pierre Delaunay, Fabrice Normandin, Mirko Bronzi, Dendi Suhubdy, Reyhane Askari, Michael Noukhovitch, Chao Xue, Satya Ortiz-Gagné, Olivier Breuleux, Arnaud Bergeron, Olexa Bilaniuk, Steven Bocco, Hadrien Bertrand, Guillaume Alain, Dmitriy Serdyuk, Peter Henderson, Pascal Lamblin, and Christopher Beckham. Epistimio/orion: Asynchronous Distributed Hyperparameter Optimization, March 2022. URL https://doi.org/10.5281/zenodo.3478592.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), Advances in Neural Information Processing Systems, volume 33, pp. 9912–9924. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/70feb62b69f16e0238f741fab228fec2-Paper.pdf.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. arXiv:2104.14294 [cs], May 2021. URL http://arxiv.org/abs/2104.14294. arXiv: 2104.14294.

Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. Adversarial robustness: From self-supervised pre-training to fine-tuning. In CVPR 2020, June 2020a.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh (eds.), Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, pp. 1597–1607. PMLR, 13–18 Jul 2020b.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. arXiv preprint arXiv:2011.10566, 2020.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3606–3613, 2014.

Adam Coates and Andrew Y. Ng. The importance of encoding versus training with sparse coding and vector quantization. In ICML, pp. 921–928, 2011. URL https://icml.cc/2011/papers/485_icmlpaper.pdf .

Gonçalo M. Correia, Vlad Niculae, and André F. T. Martins. Adaptively sparse transformers. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 2174–2184, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1223. URL https://aclanthology.org/D19-1223 .

Victor G. Turrisi da Costa, Enrico Fini, Moin Nabi, Nicu Sebe, and Elisa Ricci. Solo-learn: A library of self-supervised methods for visual representation learning, 2021. URL https://github.com/vturrisi/solo-learn .

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pp. 248–255. Ieee, 2009.

Roberto Dessì, Eugene Kharitonov, and Marco Baroni. Interpretable agent communication from scratch(with a generic visual processor emerging on the side). CoRR, abs/2106.04258, 2021. URL https://arxiv.org/abs/2106.04258 .

Roberto Dessi, Eugene Kharitonov, and Marco Baroni. Interpretable agent communication from scratch (with a generic visual processor emerging on the side). In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), Advances in Neural Information Processing Systems, 2021. URL https://openreview.net/forum?id=1AvtkM4H-y7 .

Josip Djolonga, Frances Hubis, Matthias Minderer, Zachary Nado, Jeremy Nixon, Rob Romijnders, Dustin Tran, and Mario Lucic. Robustness Metrics, 2020. URL https://github.com/google-research/robustness_metrics .

Josip Djolonga, Jessica Yung, Michael Tschannen, Rob Romijnders, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Matthias Minderer, Alexander D'Amour, Dan Moldovan, Sylvain Gelly, Neil Houlsby, Xiaohua Zhai, and Mario Lucic. On Robustness and Transferability of Convolutional Neural Networks. arXiv:2007.08558 [cs], March 2021. URL http://arxiv.org/abs/2007.08558 . arXiv: 2007.08558.

D.L. Donoho, M. Elad, and V.N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. IEEE Transactions on Information Theory, 52(1):6–18, 2006. doi: 10.1109/TIT.2005.860430.

Yann Dubois, Stefano Ermon, Tatsunori Hashimoto, and Percy Liang. Improving self-supervised learning by characterizing idealized representations. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), Advances in Neural Information Processing Systems, 2022. URL https://openreview.net/forum?id=agQGDz6gPOo .

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A. Smith. Sparse overcomplete word vector representations. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 1491–1500, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1144. URL https://aclanthology.org/P15-1144 .

Alona Fyshe, Leila Wehbe, Partha P. Talukdar, Brian Murphy, and Tom M. Mitchell. A compositional and interpretable semantic space. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 32–41, Denver, Colorado, May–June 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1004. URL https://aclanthology.org/N15-1004 .

Ian J. Goodfellow, Aaron Courville, and Yoshua Bengio. Large-scale feature learning with spike-and-slab sparse coding. In Proceedings of the 29th International Coference on International Conference on Machine Learning, ICML'12, pp. 1387–1394, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851.

Anirudh Goyal, Aniket Rajiv Didolkar, Alex Lamb, Kartikeya Badola, Nan Rosemary Ke, Nasim Rahaman, Jonathan Binas, Charles Blundell, Michael Curtis Mozer, and Yoshua Bengio. Coordination among neural modules through a shared global workspace. In International Conference on Learning Representations, 2022. URL https://openreview.net/forum?id=XzTtHjgPDsT .

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural Turing Machines. arXiv:1410.5401 [cs], December 2014. URL http://arxiv.org/abs/1410.5401 . arXiv: 1410.5401.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10, pp. 399–406, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), Advances in Neural Information Processing Systems, volume 33, pp. 21271–21284. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf .

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. Nature, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2 .

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. arXiv:1911.05722 [cs], March 2020. URL http://arxiv.org/abs/1911.05722 . arXiv: 1911.05722.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In International Conference on Learning Representations, 2019. URL https://openreview.net/forum?id=HJz6tiCqYm .

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. arXiv:2006.16241 [cs, stat], July 2021. URL http://arxiv.org/abs/2006.16241 . arXiv: 2006.16241.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In International Conference on Learning Representations, 2019. URL https://openreview.net/forum?id=Bklr3j0cKX .

Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. Neural Networks, 13:411–430, 2000.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei (eds.), Proceedings of the 32nd International Conference on Machine Learning, volume 37 of Proceedings of Machine Learning Research, pp. 448–456, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/ioffe15.html .

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In International Conference on Learning Representations, 2017. URL https://openreview.net/forum?id=rkE3y85ee .

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In International Conference on Learning Representations, 2022. URL https://openreview.net/forum?id=YevsQ05DEN7 .

M.I. Jordan and R.A. Jacobs. Hierarchical mixtures of experts and the em algorithm. In Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan), volume 2, pp. 1339–1344 vol.2, 1993. doi: 10.1109/IJCNN.1993.716791.

Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. CoRR, abs/1901.09005, 2019. URL http://arxiv.org/abs/1901.09005.

Alex Krizhevsky. Learning multiple layers of features from tiny images, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Honglak Lee, Chaitanya Ekanadham, and Andrew Ng. Sparse deep belief net model for visual area v2. In J. Platt, D. Koller, Y. Singer, and S. Roweis (eds.), Advances in Neural Information Processing Systems, volume 20. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper/2007/file/4daa3db355ef2b0e64b472968cb70f0d-Paper.pdf.

Kuang-Huei Lee, Anurag Arnab, Sergio Guadarrama, John Canny, and Ian Fischer. Compressive Visual Representations. arXiv:2109.12909 [cs, math], September 2021. URL http://arxiv.org/abs/2109.12909. arXiv: 2109.12909.

Michael S. Lewicki and Terrence J. Sejnowski. Learning Overcomplete Representations. Neural Computation, 12(2):337–365, 02 2000. ISSN 0899-7667. doi: 10.1162/089976600300015826. URL https://doi.org/10.1162/089976600300015826.

Dianbo Liu, Alex M Lamb, Kenji Kawaguchi, Anirudh Goyal ALIAS PARTH GOYAL, Chen Sun, Michael C Mozer, and Yoshua Bengio. Discrete-valued neural communication. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), Advances in Neural Information Processing Systems, volume 34, pp. 2109–2121. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/10907813b97e249163587e6246612e21-Paper.pdf.

Brian Murphy, Partha Pratim Talukdar, and Tom Michael Mitchell. Learning effective and interpretable semantic models using non-negative sparse embedding. COLING, 2012.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In 2008 Sixth Indian Conference on Computer Vision, Graphics Image Processing, pp. 722–729, 2008. doi: 10.1109/ICVGIP.2008.47.

B.A. Olshausen and D.J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature, 381:607–609, June 1996.

Bruno A. Olshausen. Highly overcomplete sparse coding. In Bernice E. Rogowitz, Thrasyvoulos N. Pappas, and Huib de Ridder (eds.), Human Vision and Electronic Imaging XVI, volume 8651 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, pp. 86510S, March 2013. doi: 10.1117/12.2013504.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural Discrete Representation Learning. arXiv:1711.00937 [cs], May 2018. URL http://arxiv.org/abs/1711.00937. arXiv: 1711.00937.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), Advances in Neural Information Processing Systems, pp. 32, 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet Classi ers Generalize to ImageNet? In Proceedings of the 36th International Conference on Machine Learning pp. 5389–5400. PMLR, May 2019. URL https://proceedings.mlr.press/v97/recht19a.html . ISSN: 2640-3498.

Aaqib Saeed, David Grangier, and Neil Zeghidour. Contrastive Learning of General-Purpose Audio Representations. arXiv:2010.10915 [cs, eess], October 2020. URL http://arxiv.org/abs/2010.10915 . arXiv: 2010.10915.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from over tting. Journal of Machine Learning Research, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html .

Yee Whye Teh, Max Welling, Simon Osindero, and Geoffrey E. Hinton. Energy-based models for sparse overcomplete representations. J. Mach. Learn. Res., 4(null):1235–1260, dec 2003. ISSN 1532-4435.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. CoRR, abs/1807.03748, 2018. URL http://arxiv.org/abs/1807.03748 .

Aad W. van der Vaart and Jon A. Wellner. Weak Convergence and Empirical Processes. Springer New York, 1996. doi: 10.1007/978-1-4757-2545-2. URL https://doi.org/10.1007%2F978-1-4757-2545-2 .

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. arXiv:1706.03762 [cs], December 2017. URL http://arxiv.org/abs/1706.03762 . arXiv: 1706.03762 version: 5.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S. Yu. Generalizing to Unseen Domains: A Survey on Domain Generalization. arXiv:2103.03097 [cs], December 2021a. URL http://arxiv.org/abs/2103.03097 . arXiv: 2103.03097.

Shulun Wang, Bin Liu, and Feng Liu. Escaping the gradient vanishing: Periodic alternatives of softmax in attention mechanism, 2021b. URL https://arxiv.org/abs/2108.07153 .

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. 2010 IEEE computer society conference on computer vision and pattern recognition, pp. 3485–3492. IEEE, 2010.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. CoRR, abs/2010.13902, 2020. URL https://arxiv.org/abs/2010.13902 .

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. arXiv preprint arXiv:2103.03230, 2021.

A PROOF OF THEOREM 1

Let us introduce additional notations used in the proofs. Define $z = (x, y) \in \mathcal{R}$, $\ell(f, r) = l(f(x), y)$,

$$C_{y;k_1,\ldots,k_L} = \{(x, \hat{y}) \in \mathcal{X} \times \mathcal{Y} : \hat{y} = y, k_j = \arg\max_{t \in [V]} x_{j,t} \quad \forall j \in [L]\},$$

and

$$\mathcal{Z}_{k_1,\ldots,k_L} = \{x \in \mathcal{X} : k_j = \arg\max_{t \in [V]} x_{j,t} \quad \forall j \in [L]\}.$$

We then define C_k to be the attention version of $C_{y;k_1,\ldots,k_L}$; i.e., $\{C_k\}_{k=1}^K = \{C_{y;k_1,\ldots,k_L}\}_{y \in \mathcal{Y}, k_1,\ldots,k_L \in [V]}$ with $C_1 = C_{1;1,\ldots,1}$, $C_2 = C_{2;1,\ldots,1}$, $C_{jYj} = C_{jYj;1,\ldots,1}$, $C_{jYj+1} = C_{1;2;1,\ldots,1}$, $C_{2jYj} = C_{jYj;2;1,\ldots,1}$, and so on. Similarly, define $\mathcal{Z}_k$ to be the attention version of $\mathcal{Z}_{k_1,\ldots,k_L}$. We also use $Q_i = \{q \in [-1, +1]^V : i = \arg\max_{j \in [V]} q_j\}$, $I_k := I_k^S := \{i \in [n] : r_i \in C_k\}$, and $\pi_k(h) := E_r[\ell(h, r) | r \in C_k]$. Moreover, we define $\Psi(f_{\text{base}}^S) = \sup_{i \in [V]} \sup_{q,q' \in Q_i} \|q - q'\|_2^2$, and $\Psi(f_{\text{SEM}(\cdot)}^S) = \sup_{i \in [V]} \sup_{q,q' \in Q_i} \|\sigma(q) - \sigma(q')\|_2^2$ where $\sigma(q)_j = \frac{e^{q_j}}{\sum_{t=1}^V e^{q_t}}$ for $j = 1, \ldots, V$.

We first decompose the generalization gap into two terms using the following lemma:

Lemma 1. For any $\epsilon > 0$, with probability at least $1 - \epsilon$, the following holds for all $h \in H$:

$$E_r[\ell(h, r)] - \frac{1}{n}\sum_{i=1}^n \ell(h, r_i) \le \sum_{k=1}^K \frac{|I_k|}{n}\left(\pi_k(h) - \frac{1}{|I_k|}\sum_{i \in I_k} \ell(h, r_i)\right) + c\sqrt{\frac{\ln(2/\epsilon)}{n}}.$$

Proof. We first write the expected error as the sum of the conditional expected error:

$$E_r[\ell(h, r)] = \sum_{k=1}^K E_r[\ell(h, r) | r \in C_k] \Pr(r \in C_k) = \sum_{k=1}^K E_{r_k}[\ell(h, r_k)] \Pr(r \in C_k),$$

where r_k is the random variable for the conditional with $r \in C_k$. Using this, we decompose the generalization error into two terms:

$$\begin{aligned} &E_r[\ell(h, r)] - \frac{1}{n}\sum_{i=1}^n \ell(h, r_i) \qquad (3)\\ &= \sum_{k=1}^K E_{r_k}[\ell(h, r_k)] \Pr(r \in C_k) - \frac{|I_k|}{n} + \left(\sum_{k=1}^K E_{r_k}[\ell(h, r_k)]\frac{|I_k|}{n} - \frac{1}{n}\sum_{i=1}^n \ell(h, r_i)\right). \end{aligned}$$

The second term in the right-hand side of (3) is further simplified by using

$$\frac{1}{n}\sum_{i=1}^n \ell(h, r_i) = \frac{1}{n}\sum_{k=1}^K \sum_{i \in I_k} \ell(h, r_i),$$

as

$$\sum_{k=1}^K E_{r_k}[\ell(h, r_k)]\frac{|I_k|}{n} - \frac{1}{n}\sum_{i=1}^n \ell(h, r_i) = \frac{1}{n}\sum_{k=1}^K |I_k|\left(E_{r_k}[\ell(h, r_k)] - \frac{1}{|I_k|}\sum_{i \in I_k} \ell(h, r_i)\right)$$

Substituting these into equation (3) yields

$$\begin{aligned} &E_r[\ell(h, r)] - \frac{1}{n}\sum_{i=1}^n \ell(h, r_i) \qquad (4)\\ &= \sum_{k=1}^K E_{r_k}[\ell(h, r_k)] \Pr(r \in C_k) - \frac{|I_k|}{n} + \frac{1}{n}\sum_{k=1}^K |I_k|\left(E_{r_k}[\ell(h, r_k)] - \frac{1}{|I_k|}\sum_{i \in I_k} \ell(h, r_i)\right)\\ &\le \left(\sum_{k=1}^K \Pr(r \in C_k) - \frac{|I_k|}{n} + \frac{1}{n}\sum_{k=1}^K |I_k|\left(E_{r_k}[\ell(h, r_k)] - \frac{1}{|I_k|}\sum_{i \in I_k} \ell(h, r_i)\right)\right) \end{aligned}$$

By using the Bretagnolle-Huber-Carol inequality (van der Vaart & Wellner, 1996, A6.6 Proposition), we have that for any > 0, with probability at least 1 r,

$$\sum_{k=1}^{K} \Pr(r \in C_k) \left| \frac{|I_k|}{n} \right| \sqrt{\frac{2K \ln(2/)}{n}}. \quad (5)$$

Here, notice that the term of $\sum_{k=1}^{K} \Pr(r \in C_k) \left| \frac{|I_k|}{n} \right|$ does not depend on $h \in H$. Moreover, note that for any $(f; h; M)$ such that $M > 0$ and $B \geq 0$ for all X, we have that $P(f(X) \geq M) \leq P(f(X) > M) \leq P(Bf(X) + h(X) > BM + h(X))$; where the probability is with respect to the randomness of X. Thus, by combining (4) and (5), we have that for any $h \in H$, for any > 0, with probability at least 1 , the following holds for all $h \in H$,

$$E_r[\ell(h; r)] \leq \frac{1}{n} \sum_{i=1}^{n} \ell(h; r_i) \leq \frac{1}{n} \sum_{k=1}^{K} |I_k| \left(\Delta_k(h) + \frac{1}{|I_k|} \sum_{i \in I_k} \ell(h; r_i) \right) + c \sqrt{\frac{\ln(2/)}{n}}.$$

□

In particular, the first term from the previous lemma will be bounded with the following lemma:

Lemma 2. For any $f \in \{f^S_{SEM(\ )}, f^S_{base}\}$,

$$\frac{1}{n} \sum_{k=1}^{K} |I_k| \left(\Delta_k(f) + \frac{1}{|I_k|} \sum_{i \in I_k} \ell(f; r_i) \right) \leq R \sqrt{L'(f)}.$$

Proof. By using the triangle inequality,

$$\frac{1}{n} \sum_{k=1}^{K} |I_k| \left(\Delta_k(f) + E_r[\ell(f; r)|r \in C_k] - \frac{1}{|I_k|} \sum_{i \in I_k} \ell(f; r_i) \right)$$

$$\leq \frac{1}{n} \sum_{k=1}^{K} |I_k| \left(\Delta_k(f) + E_r[\ell(f; r)|r \in C_k] - \frac{1}{|I_k|} \sum_{i \in I_k} \ell(f; r_i) \right) :$$

Furthermore, by using the triangle inequality,

$$E_r[\ell(f; r)|r \in C_k] - \frac{1}{|I_k|} \sum_{i \in I_k} \ell(f; r_i) = \frac{1}{|I_k|} \sum_{i \in I_k} E_r[\ell(f; r)|r \in C_k] - \frac{1}{|I_k|} \sum_{i \in I_k} \ell(f; r_i)$$

$$\leq \frac{1}{|I_k|} \sum_{i \in I_k} E_r[\ell(f; r)|r \in C_k] - \ell(f; r_i)$$

$$\leq \sup_{r, r' \in C_k} \ell(f; r) - \ell(f; r') :$$

If $f = f^S_{SEM(\ )} = g^S_{SEM(\ )}$, since $g^S_{SEM(\ )} \in G_S$, by using the Lipschitz continuity, boundedness, and non-negativity,

$$\sup_{r, r' \in C_k} \ell(f; r) - \ell(f; r') = \sup_{y \in Y} \sup_{z, z' \in Z_k} |(l_y \circ g^S_{SEM(\ )})(\ (z)) - (l_y \circ g^S_{SEM(\ )})(\ (z'))|$$

$$\leq R \sup_{z, z' \in Z_k} \| \ (z) - \ (z') \|_F$$

$$= R \sup_{z, z' \in Z_k} \sqrt{\sum_{t=1}^{} \sum_{j=1}^{} (\ (z_{t;j}) - \ (z'_{t;j}))^2}$$

$$\leq R \sqrt{\sum_{t=1}^{} \sup_{i \in [V]} \sup_{q, q' \in Q_i} \| \ (q) - \ (q') \|_2^2}$$

$$= R \sqrt{L'(f^S_{SEM(\ )})}$$

Similarly, if $f = f^S_{\text{base}} = g^S_{\text{base}}$, since $g^S_{\text{base}} \in G_S$, by using the Lipschitz continuity, boundedness, and non-negativity,

$$\sup_{r,r' \in C_k} \ell(f,r) - \ell(f,r') = \sup_{y \in Y} \sup_{z,z' \in Z_k} |(l_y - g^S_{\text{base}})(z) - (l_y - g^S_{\text{base}})(z')|$$
$$\leq R \sup_{z,z' \in Z_k} \|z - z'\|_F \sqrt{L'(f^S_{\text{base}})}.$$

Therefore, for any $f \in \{f^S_{\text{SEM}(\cdot)}, f^S_{\text{base}}\}$,

$$\frac{1}{n}\sum_{k=1}^{K} |I_k| \left(\epsilon_k(f) - \frac{1}{|I_k|} \sum_{i \in I_k} \ell(f, r_i) \right) + \frac{1}{n}\sum_{k=1}^{K} |I_k| R \sqrt{L'(f)} = R \sqrt{L'(f)}.$$

□

Combining Lemma 1 and Lemma 2, we obtain the following upper bound on the gap:

Lemma 3. For any $\delta > 0$, with probability at least $1 - \delta$, the following holds for any $f \in \{f^S_{\text{SEM}(\cdot)}, f^S_{\text{base}}\}$:

$$E_r[\ell(f,r)] - \frac{1}{n}\sum_{i=1}^{n} \ell(f,r_i) \leq R \sqrt{L'(f)} + c \sqrt{\frac{\ln(2/\delta)}{n}}.$$

Proof. This follows directly from combining Lemma 1 and Lemma 2. □

We now provide an upper bound on $\psi(f^S_{\text{SEM}(\cdot)})$ in the following lemma:

Lemma 4. For any $\beta > 0$,

$$\psi(f^S_{\text{SEM}(\beta)}) \leq \left(\frac{1}{1 + (V-1)e^{-2/\beta}} - \frac{1}{1 + (V-1)e^{-1/\beta}} \right)^2$$
$$+ (V-1) \left(\frac{1}{1 + e^{-1/\beta}(1 + (V-2)e^{-2/\beta})} - \frac{1}{1 + e^{-2/\beta}(1 + (V-2)e^{-1/\beta})} \right)^2.$$

Proof. Recall the definition:

$$\psi(f^S_{\text{SEM}(\beta)}) = \sup_{i \in [V]} \sup_{q,q' \in Q_i} \|\sigma(q) - \sigma(q')\|_2^2.$$

where

$$\sigma(q)_j = \frac{e^{q_j/\beta}}{\sum_{t=1}^{V} e^{q_t/\beta}},$$

for $j = 1, \ldots, V$. By the symmetry and independence over $i \in [V]$ inside of the first supremum, we have

$$\psi(f^S_{\text{SEM}(\beta)}) = \sup_{q,q' \in Q_1} \|\sigma(q) - \sigma(q')\|_2^2.$$

For any $q, q' \in Q_1$ and $i \in \{2, \ldots, V\}$ (with $q = (q_1, \ldots, q_V)$ and $q' = (q'_1, \ldots, q'_V)$), there exists $\alpha_i, \alpha'_i > 0$ such that

$$q_i = q_1 - \alpha_i$$

and

$$q'_i = q'_1 - \alpha'_i.$$

Here, since $z_{ik} \neq z_{ij}$ from the assumption, we have that for all $i \in \{2, \ldots, V\}$,

$$\alpha_i, \alpha'_i > 0.$$

Thus, we can rewrite

$$\begin{aligned}\sum_{t=1}^{V} e^{q_t/\lambda} &= e^{q_1/\lambda} + \sum_{i=2}^{V} e^{(q_1 - \varepsilon_i)/\lambda} \\ &= e^{q_1/\lambda} + e^{q_1/\lambda} \sum_{i=2}^{V} e^{-\varepsilon_i/\lambda} \\ &= e^{q_1/\lambda} \left(1 + \sum_{i=2}^{V} e^{-\varepsilon_i/\lambda}\right)\end{aligned}$$

Similarly,

$$\sum_{t=1}^{V} e^{q_t^0/\lambda} = e^{q_1^0/\lambda} \left(1 + \sum_{i=2}^{V} e^{-\varepsilon_i^0/\lambda}\right):$$

Using these,

$$(q)_1 = \frac{e^{q_1/\lambda}}{\sum_{t=1}^{V} e^{q_t/\lambda}} = \frac{e^{q_1/\lambda}}{e^{q_1/\lambda}\left(1 + \sum_{i=2}^{V} e^{-\varepsilon_i/\lambda}\right)} = \frac{1}{1 + \sum_{i=2}^{V} e^{-\varepsilon_i/\lambda}}$$

and for all$j \in \{2, \ldots, V\}$,

$$\begin{aligned}(q)_j &= \frac{e^{q_j/\lambda}}{\sum_{t=1}^{V} e^{q_t/\lambda}} \\ &= \frac{e^{(q_1 - \varepsilon_j)/\lambda}}{e^{q_1/\lambda}\left(1 + \sum_{i=2}^{V} e^{-\varepsilon_i/\lambda}\right)} \\ &= \frac{e^{-\varepsilon_j/\lambda}}{1 + \sum_{i=2}^{V} e^{-\varepsilon_i/\lambda}} \\ &= \frac{1}{1 + e^{\varepsilon_j/\lambda} + \sum_{i \in I_j} e^{(\varepsilon_j - \varepsilon_i)/\lambda}}\end{aligned}$$

where$I_j := \{2, \ldots, V\} \setminus \{j\}$. Similarly,

$$(q^0)_1 = \frac{1}{1 + \sum_{i=2}^{V} e^{-\varepsilon_i^0/\lambda}};$$

and for all$j \in \{2, \ldots, V\}$,

$$(q^0)_j = \frac{1}{1 + e^{\varepsilon_j^0/\lambda} + \sum_{i \in I_j} e^{(\varepsilon_j^0 - \varepsilon_i^0)/\lambda}}:$$

Using these, for any$q, q^0 \in Q_1$,

$$\begin{aligned}|(q)_1 - (q^0)_1| &= \left|\frac{1}{1 + \sum_{i=2}^{V} e^{-\varepsilon_i/\lambda}} - \frac{1}{1 + \sum_{i=2}^{V} e^{-\varepsilon_i^0/\lambda}}\right| \\ &\quad \left|\frac{1}{1 + \sum_{i=2}^{V} e^{-\varepsilon_i/\lambda}} - \frac{1}{1 + \sum_{i=2}^{V} e^{-\varepsilon_i^0/\lambda}}\right| \\ &= \frac{1}{1 + (V-1)e^{-2/\lambda}} \cdot \frac{1}{1 + (V-1)e^{-/\lambda}};\end{aligned}$$

and for all $j \in \{2, \ldots, V\}$,

$$\begin{aligned}
j\ (q)_j - (q^0)_j\ j &= \left| \frac{1}{1 + e^{-\beta_j} + \sum_{i \neq 1, j} e^{-(\alpha_j - \alpha_i)/\tau}} - \frac{1}{1 + e^{-\beta_j^0} + \sum_{i \neq 1, j} e^{-(\alpha_j^0 - \alpha_i^0)/\tau}} \right| \\
&\quad \left| \frac{1}{1 + e^{-\beta_2/\tau} + \sum_{i \neq 1, j} e^{-(\alpha_2 - \alpha_i)/\tau}} - \frac{1}{1 + e^{-\beta_2^0/\tau} + \sum_{i \neq 1, j} e^{-(\alpha_2^0 - \alpha_i^0)/\tau}} \right| \\
&= \left| \frac{1}{1 + e^{-\beta_j/\tau} + (V-2)e^{-(\alpha_j - \alpha_2)/\tau}} - \frac{1}{1 + e^{-\beta_j^0/\tau} + (V-2)e^{-(\alpha_j^0 - \alpha_2^0)/\tau}} \right| \\
&\quad \left| \frac{1}{1 + e^{-\beta_2/\tau} + (V-2)e^{-(\alpha_2 - \alpha_j)/\tau}} - \frac{1}{1 + e^{-\beta_2^0/\tau} + (V-2)e^{-(\alpha_2^0 - \alpha_j^0)/\tau}} \right| \\
&= \left| \frac{1}{1 + e^{-\beta_j/\tau} (1 + (V-2)e^{-\alpha_2/\tau})} - \frac{1}{1 + e^{-\beta_j^0/\tau} (1 + (V-2)e^{-\alpha_2^0/\tau})} \right| \\
&\quad \left| \frac{1}{1 + e^{-\beta_2/\tau} (1 + (V-2)e^{-\alpha_j/\tau})} - \frac{1}{1 + e^{-\beta_2^0/\tau} (1 + (V-2)e^{-\alpha_j^0/\tau})} \right| .
\end{aligned}$$

By combining these,

$$\begin{aligned}
&\sup_{q, q^0 \in Q_1} \| (q) - (q^0) \|_2^2 \\
&= \sup_{q, q^0 \in Q_1} \sum_{j=1}^{V} j\ (q)_j - (q^0)_j\ j^2 \\
&\quad \left| \frac{1}{1 + (V-1)e^{-\beta_2/\tau}} - \frac{1}{1 + (V-1)e^{-\beta_2^0/\tau}} \right|^2 \\
&\quad + (V-1) \left| \frac{1}{1 + e^{-\beta_j/\tau} (1 + (V-2)e^{-\alpha_2/\tau})} - \frac{1}{1 + e^{-\beta_j^0/\tau} (1 + (V-2)e^{-\alpha_2^0/\tau})} \right|^2 .
\end{aligned}$$

□

Using the previous lemma, we will conclude the asymptotic behavior of $\Delta(f^S_{SEM(\ )})$ in the following lemma:

Lemma 5. It holds that

$$\Delta(f^S_{SEM(\ )}) \to 0 \text{ as } \to 0.$$

Proof. Using Lemma 4,

$$\begin{aligned}
\lim_{\to 0} \Delta(f^S_{SEM(\ )}) &\leq \lim_{\to 0} \left| \frac{1}{1 + (V-1)e^{-\beta_2/\tau}} - \frac{1}{1 + (V-1)e^{-\beta_2^0/\tau}} \right|^2 \\
&\quad + n(V-1) \lim_{\to 0} \left| \frac{1}{1 + e^{-\beta_j/\tau} (1 + (V-2)e^{-\alpha_2/\tau})} - \frac{1}{1 + e^{-\beta_2^0/\tau} (1 + (V-2)e^{-\alpha_2^0/\tau})} \right|^2 .
\end{aligned}$$

Moreover,

$$\lim_{\to 0} \left| \frac{1}{1 + (V-1)e^{-\beta_2/\tau}} - \frac{1}{1 + (V-1)e^{-\beta_2^0/\tau}} \right|^2 = \left| \frac{1}{1} - \frac{1}{1} \right|^2 = 0;$$

and

$$\lim_{\to 0} \left| \frac{1}{1 + e^{-\beta_j/\tau} (1 + (V-2)e^{-\alpha_2/\tau})} - \frac{1}{1 + e^{-\beta_2^0/\tau} (1 + (V-2)e^{-\alpha_2^0/\tau})} \right|^2 = |0 - 0|^2 = 0.$$

Therefore,

$$\lim_{\to 0} \Delta(f^S_{SEM(\ )}) \leq 0.$$

Since $\Delta(f^S_{SEM(\ )}) \geq 0$, this implies the statement of this lemma. □

As we have analyzed $\Delta(f^S_{SEM(\ )})$ in the previous two lemmas, we are now ready to compare $\Delta(f^S_{SEM(\ )})$ and $\Delta(f^S_{base})$, which is done in the following lemma:

Lemma 6. For any $\ > 0$,

$$\Delta(f^S_{SEM(\ )}) \quad \Delta(f^S_{base}) \quad \frac{3}{4}(1 \quad V) < 0:$$

Proof. From Lemma 4, for any > 0,

$$\begin{aligned}
\Delta(f^S_{SEM(\ )}) &\quad \frac{1}{1+(V \quad 1)e^{\ 2=}} \quad \frac{1}{1+(V \quad 1)e^{\ =}}^2 \\
&\quad + n(V \quad 1)\ \frac{1}{1+e^{\ =}(1+(V \quad 2)e^{\ 2=})} \quad \frac{1}{1+e^{2=}(1+(V \quad 2)e^{\ =})}^2 \\
&\quad \frac{1}{1+(V \quad 1)e^{\ 2=}} \quad \frac{1}{1+(V \quad 1)}^2 \\
&\quad + (V \quad 1)\ \frac{1}{1+(1+(V \quad 2)e^{\ 2=})} \quad \frac{1}{1+e^{2=}(1+(V \quad 2))}^2 \\
= &\quad \frac{1}{1+(V \quad 1)e^{\ 2=}} \quad \frac{1}{V}^2 + (V \quad 1)\ \frac{1}{2+(V \quad 2)e^{\ 2=}} \quad \frac{1}{1+e^{2=}(V \quad 1)}^2 \\
&\quad \frac{1}{1} \quad \frac{1}{V}^2 + (V \quad 1)\ \frac{1}{2} \quad 0^2 \\
= &\quad \frac{1}{1} \quad \frac{1}{V}^2 + (V \quad 1)\frac{1}{4}:
\end{aligned}$$

Recall the de nition of

$$\Delta(f^S_{base}) = \sup_{i\,2\,[V]}\ \sup_{q;q^0 2\,Q_i} kq \quad q^0k_2^2:$$

By choosing an element in the set over which the supremum is taken, for any > 0,

$$\Delta(f^S_{base}) \quad \sup_{q;q^0 2\,Q_1} kq \quad q^0k_2^2 \quad k\ \hat{q}_1 \quad \hat{q}_1^0k_2^2 = \sum_{j=1}^{V}(\hat{q}_j \quad \hat{q}_j^0)_2^2 = (2 \quad)^2V;$$

where $\hat{q}_1 = 1$, $\hat{q}_j = 1$ for $j\ 2\ f\ 2;:::;Vg$, $\hat{q}_1^0 = \quad 1$, and $\hat{q}_j^0 = \quad 1$ for $j\ 2\ f\ 2;:::;Vg$.

By combining those, for for any > 0 and > 0,

$$\begin{aligned}
\Delta(f^S_{SEM(\ )}) \quad \Delta(f^S_{base}) &\quad \frac{1}{1} \quad \frac{1}{V}^2 + (V \quad 1)\frac{1}{4} \quad (2 \quad)^2V \\
&\quad 1 + \frac{1}{4}V \quad \frac{1}{4} \quad (2 \quad)^2V \\
&= \frac{3}{4} + \frac{1}{4}V \quad (2 \quad)^2V \\
&= \frac{3}{4} \quad V \quad (2 \quad)^2 \quad \frac{1}{4} \\
&\quad \frac{3}{4} \quad V \quad 1 \quad \frac{1}{4} \\
&= \frac{3}{4}(1 \quad V)
\end{aligned}$$

□

We combine the lemmas above to prove Theorem 1, which is restated below with its proof:

Theorem 1. Let V 2. For any 1 $>$ 0, with probability at least 1 , the following holds for any f_S 2 $f f^S_{SEM(\ )}; f^S_{base} g$:

$$E_{z;y}[l(f_S(z); y)] \quad \frac{1}{n}\sum_{i=1}^{n} l(f_S(z^{(i)}); y^{(i)}) + R \sqrt{\frac{L' f_S(V; \ )}{} + c \sqrt{\frac{\ln(2=\ )}{n}}};$$

where $c > 0$ is a constant in $(n; f; H; \ ; S)$. Moreover,

$$' f^S_{SEM(\ )} ! \ 0 \ \text{as} \ ! \ 0 \ \text{and} \ ' f^S_{SEM(\ )} \quad ' f^S_{base} \quad \frac{3}{4}(1 \quad V) < 0 \ \ 8 > 0:$$

Proof. The first statement directly follows from Lemma 3. The second statement is proven by Lemma 5 and Lemma 6. □

B EXPERIMENT DETAILS FOR IMAGENET

B.1 IMAGE AUGMENTATION

The augmentation applied in order during training are:

- Random Resize crop to 224 224 image. A random patch of the image is selected and resized to a 224 224 image.
- Random color jitter. Modifying the brightness, the contrast, the saturation and the hue.
- Random gray scale. Randomly applying a gray scale filter to the image
- Random Gaussian blur. Randomly applying a Gaussian bluer filter.
- Random solarization. Randomly applying a solarization filter.

The parameters of the augmentations are presented in Table 16. At validation and test time, we resize the images to 256 256 and then center crop a patch of 224 224.

For both training and evaluation, we re-normalize the image using the statistic of the training set.g

B.2 LINEAR EVALUATION

We follow the evaluation protocol from (Chen et al., 2020b). The linear evaluation is done by training a linear classifier on the frozen representation of the ImageNet training samples. We train a linear classifier with a cross-entropy objective for 100 epochs using SGD with nesterov, a momentum of 0:9 and a batch size of 256. We perform learning rate scheduling at epoch 60 and epoch 80 where we divide the learning rate by a factor of 10. During training, we apply random resized crop to 224 224 pixels and random horizontal flip. We sweep over a set of 4 learning rates $f 0:5; 0:1; 0:05; 0:01g$, 3 l1 weight decays $f 0; 1e \quad 6; 1e \quad 5g$ and 3 $_d$ for SEM: $f 0:01; 0:1; 1g$, using a validation set of 10 images per class and re-traing using the full training set. We report the results on the test set.

B.3 ROBUSTNESS EXPERIMENTS

We follow the evaluation procedure from (Lee et al., 2021). We treated the robustness datasets as additional "test sets" in that we simply evaluated them using the evaluation procedure described above. The images were resized to 256 256 before being center cropped to 224 224 image.
The evaluation procedure was performed using the public robustness benchmark evaluation code of (Djolonga et al., 2020[§])

B.4 TRANSFER LEARNING LINEAR PROBE

We follow the linear evaluation protocol of (Kolesnikov et al., 2019; Chen et al., 2020b) We train a linear classifier using a regularized multinomial logistic regression from the scikit-learn package (Pedregosa et al., 2011). The representation is frozen, so that we do not train the encoder backbone nor

[§] https://github.com/google-research/robustness_metrics

the batch-normalization statistics. We do not perform any augmentations and the images are resized to 224 pixels using bicubic resampling and the normalized using the statistics on ImageNet's training set. We tune the regularizer term from a range of 45 logarithmically-spaced values between 10^{-6} and 10^{5} using a small validation set and re-train using the full training set. For SEM, we set $\alpha = 0$ for all experiments.

B.5 TRANSFER LEARNING FINE-TUNING

We follow the same fine-tuning protocol of (Chen et al., 2020b; Grill et al., 2020). We initialize the encoder with the pre-trained model and a classifier head with random initialization. We train for 20,000 steps with a batch size of 256 using SGD with a Nesterov momentum of 0.9. We set the momentum parameter for the batch normalization to $\max(1 - 10/s, 0.9)$ where s is the number of steps per epoch. During pre-training, we use random resize crop to 224×224 pixels and random horizontal flipping. At test time, we resize the images along the shortest side to 256 pixels using cubic resampling following by a center resize crop to 224×224 pixels. Due to computational constraint, we only tune the learning rate using a search of 7 values spaces on logarithmic scales between 0.0001 and 0.1. For SEM, we set $\alpha = 1.$ for all experiments After choosing the best learning rate of a validation set, we re-run the models using the full training set and evaluate it on the test set, which we use to report the numbers.

B.6 SEMI-SUPERVISED LEARNING

We follow the semi-supervised learning protocol of (Chen et al., 2020b; Grill et al., 2020). We initialize the network using the pre-trained representation and initialize a classification head using random initialization. We fine-tune the encoder while training the classification head using a small subset of ImageNet. We choose the same subset used in prior works which is defined in the TensorFlow-Dataset software. During training, we random resize the images to 224×224 pixels along the shorter size using bicubic resampling followed by a center crop and random horizontal flipping. At test time, we resize the image to 224×224. We optimize the cross entropy loss with nestorov and a momentum of 0.9 using batch sizes of 1024. We train models for $\{30, 50\}$ and take the best performing on the validation set. The learning rate used is chosen among a search of 5 learning rates: $\{0.01, 0.02, 0.05, 0.1, 0.005\}$. For SEM, we also search $\alpha \in \{0.01, 0.1, 1\}$. We perform the search on the best performing one on the validation set and the number are returned are obtained using the test set after re-training using the full training set.

C HYPERPARAMETERS

The implementation of the SSL methods used in this work are taken from Solo-Learn (da Costa et al., 2021) to which we added the SEM module. The pre-training hyper-parameters of every SSL methods trained on CIFAR-100 with ResNet-18 used in this work are the default provided in the companion repository of Solo-Learn. The hyper-parameters are also provided in the launch scripts accompanying this work. Due to the large number of SSL methods probed in this work and the amount of space it would require to exhaustively detail all of the hyper-parameters, we refer the reader to the code.

For the CIFAR-100 results obtained with BYOL and a ResNet-50, we have slightly modified the default parameters. Otherwise, the baseline BYOL model would not obtain competitive results. The hyper-parameters were tuned using the BYOL baseline and the SEM module was not considered in the selection of the SSL hyper-parameters. The BYOL hyper-parameters are presented in the launch script accompanying this work and presented below for completeness.

For the ImageNet experiments, we took the hyper-parameters proposed in the launch scripts of Solo-Learn to which we only modified the amount of epochs (100 epochs to 200 epochs.)

Here, we present all of the SEM hyper-parameters used in every experiments. These hyper-parameters can also be found in the launch scripts accompanying this work.

We present the hype-parameters used to train for BYOL+SEM and MoCo+SEM on CIFAR100. Unless mentioned otherwise, these are the parameters used.

Table 6: BYOL with ResNet-50 for CIFAR-100.

precision	16
Learning rate	0.5
Weight-decay	1e-4
Optimizer	sgd + lars
LR scheduler	warmup + cosine
eta lars	0.001
exclude bias n norm (lars)	True
batch size	256
base ema momentum	0.99
nal ema momentum	1.0
proj output dim	256
proj hidden dim	4096
pred hidden dim	4096
augmentations	
solarization_prob	view 1: 0 view 2: 0.2
crop size	32
hue	0.1
saturation	0.2
contrast	0.4
brightness	0.4

Table 7: SEM SimCLR RN-18 for CIFAR-100

L	V	p	0 p
5000	13	0.17	0.78

Table 8: SEM MoCo RN-18 for CIFAR-100

L	V	p	0 p
5000	13	0.04	0.01

Table 9: SEM BYOL RN-18 for CIFAR-100

L	V	p	
5000	13	10	1.0

Table 10: SEM SwAV RN-18 for CIFAR-100

L	V	p	0 p
5000	13	0.85	1.5

Table 11: SEM DINO RN-18 for CIFAR-100

L	V	p	0 p
5000	13	10	1.0

Table 12: SEM Barlow RN-18 for CIFAR-100

L	V	p	0 p
5000	13	10	0.99

Table 13: SEM VicREG RN-18 for CIFAR-100

L	V	p	0 p
5000	13	10	1.0

Table 14: SEM BYOL RN-50 for CIFAR-100

L	V	p	0 p
5000	13	1	1

Table 15: SEM BYOL all ResNets for ImageNet

L	V	p	0 p
5000	21	0.16	0.04

C.1 Computational resources

For all our CIFAR-100 training, we used 1 RTX-8000 per experiment. For our ImageNet experiments, we used parallel training with 2 40GB A100 for the training with ResNet50 and ResNet50-x2 and 4 40GB A100 for the training with ResNet50-x4. With this setup, the training takes about a week for the ResNet50 experiments and about 10 days for the ResNet50-x2 and ResNet50-x4 experiments.

D Additional studies of SEM

In Section 4.2, we discussed the effect of scaling L and V as well as changing the Softmax temperature during pre-training of the online network and changing the Softmax temperature for the downstream task. Here, we propose additional studies of SEM to provide a better mastery of the method. We provide a method for reducing the memory overhead of SEM and experiments demonstrating that despite this version still largely outperform the baseline. We additionally present the effect of modifying the embedder contributing to the insight on how to get the most out of SEM. Next, we have discussion with a study of the spectrum of the covariance matrix of the SEM representation and the BYOL representation, showing insight how SEM can particularly improve the training signal

Table 16: BYOL with all ResNet-50 architectures for ImageNet.

precision	16
Learning rate	0.4
Weight-decay	1e-6
Optimizer	sgd + lars
LR scheduler	warmup + cosine
eta lars	0.001
exclude bias n norm (lars)	True
batch size	256
base ema momentum	0.99
nal ema momentum	1.0
proj output dim	256
proj hidden dim	4096
pred hidden dim	4096
augmentations	
solarization_prob	view 1: 0 view 2: 0.2
gaussian_prob	view 1: 1.0 view 2: 0.1
crop size	224
hue	0.1
saturation	0.2
contrast	0.4
brightness	0.4

Table 17: # of parameters, # of activations, allocated memory, computation ef ciency (FLOPs/sample) and CIFAR-100 accuracy of BYOL, BYOL with SEM and its memory-ef cient variant with 8 blocks (denoted BYOL + SEM/8).

	# params	# activations	vRAM (GiB)	FLOPs	Accuracy
Resnet-18					
BYOL	16.5M	0.731M	4.0	7.20e8	70.7
BYOL+SEM	313.7M	0.797M	13.1	1.01e9	73.9
BYOL+SEM/8	51.9M	0.796M	5.3	7.46e8	73.3
Resnet-50					
BYOL	35M	4.05M	11.1	1.65e9	74.3
BYOL+SEM	425.6M	4.12M	21.9	2.04e9	77.4
BYOL+SEM/8	76.7M	4.12M	11.8	1.69e9	76.6

during pre-training. We provide a scaling analysis of BYOL and BYOL + SEM on CIFAR-100. We end with an experiment showing that pre-training with SEM is necessary to get the best performance.

D.1 An efficient variant of SEM

A large over-complete representation may induce a signi cant memory footprint due to the additional parameters of the fully connected linear layer used to map to and from the representation. For SEM we require two such mappings as depicted in Figure 2c for BYOL. To reduce the amount of parameters, we propose to sparsify the weight matrix of the fully connected linear layer. We propose to do so by taking the block diagonal of the parameters of the matrix multiplication and setting the parameters outside the block diagonal to 0. Formally, let$v \in \mathbb{R}^{b \times m}$, $w \in \mathbb{R}^{m \times o}$ and$y = v \cdot w$ be the fully connected matrix multiplication. Instead, we partition$v$ into n blocks with$v^i \in \mathbb{R}^{b \times \frac{m}{n}}$ and de ne smaller$w^i \in \mathbb{R}^{\frac{m}{n} \times \frac{o}{n}}$, where$i \in [L]$ is thei^{th} block. Then, we perform a batch matrix multiplication of v^i andw^i that we concatenate as follows:$y^i = v^i \cdot w^i$ and$y^i = \text{Concat}([y^1, \dots, y^n])$. Thus, the amount of parameters of this matrix multiplication scales in$\mathcal{O}(\frac{m \cdot o}{n})$, allowing us to reduce the memory consumption by increasing$n$ the number of blocks.

We perform an experiment where we partition the embedder and the rst linear layer of the projector into 8 blocks. We present the results in Table 17 in which we compare the # of parameters, the

# of activations, the allocated vRAM by pytorch, the FLOPs/sample and the accuracy of BYOL, BYOL+SEM and BYOL+SEM/8 representing the model with $\frac{1}{8}$th blocks obtained following the method described above. We observe that partitioning the matrix multiplications of SEM allows to vastly reduce the computation parameters while still yielding an important improvement over the baseline. This result demosntrate that SEM can be bene cial while inducing minimal computational overhead.

Attentive readers may notice that this performance is better compared to the ablation presented in Figure 3. The difference in performance is due to probing the embedder's output (z) in Figure 3 and probing the encoder's output (i.e.) in Table 17. Using the each ablation's representation for probing to the other recovers the performance observed by each.

D.2 Additional ablation of the SEM parameters

Ablating the embedder In the main text, we mentioned that we use batch normalization)) at the output of the embedder. The reason we use batch normalization is mostly due to the fact that we wanted to avoid tuning any hyper-parameters that were not related to SEM to emphasize its contribution. Using BatchNorm gave the best performance without tuning the hyper-parameters of the baseline models.

Here, we want to emphasize that SEM can be used without batch norm, but more hyper-parameters might need to be tuned for it to perform as well as the model with batch norm in the encoder. For example, we found that using no weight decay was important to get better performance when we did not have batch normalization as illustrated in Table table 18. We leave the full study of the interaction of SEM with the SSL related parameters for future work.

Table 18: Understanding the relationship between the use of BatchNorm in the embedder and the weight decay hyper-parameter.

BatchNorm	weight decay	Accuracy
	0	67.2
	1e-5	57.9
X	0	68.3
X	1e-5	73.9

Another decision is to use a linear layer as the embedder. Other alternative may include using the Identidy function (i.e. the output of the encoder is used for SEM). However, if we want to systematically use the same encoder as the SSL model, then we are constrained to a representation size that is the one of the ResNet encoder (i.e. 512 for a ResNet-18).

Finally, we showcase that using a more expressive embedder leads to exacerbated performance and recommend practitioner to limit the expressivity of their embedder.

Table 19: Comparing alternative embedders.

	Accuracy
Identity	63.0
Linear	73.9
1 hidden layer MLP	65.0

A very very large embedding Using a ResNet-18 encoder and the method proposed in Section D.1, we further scale the embedding size of SEM to see where the performance saturates for classi cation. In Figure 7 we observe that the performance saturates for 10000 for the classi cation task. We conjecture that the optimal might be different for other tasks, but we leave that study for future work.

D.3 Analyze of the spectrum of the covariance matrix of the representation

To obtain a better insight on why the SEM representation leads to better downstream performance, we analyze the spectrum of the covariance matrix of the representation using the methodology presented

Figure 7: Study of very very large using a ResNet-18 backbone and SEM/8 blocks using the method described in Section D.1.

in Jing et al. (2022). That is, we collect the embedding vectors of the test set of CIFAR-100 using a pre-trained model using ResNet-50. For BYOL, we have an additional embedder without softmax normalization (as done in Figure 3). For BYOL and BYOL+SEM we use the embedder's output (z) to perform the evaluation. To compute the covariance matrix $C \in \mathbb{R}^{L \times V \times L \times V}$ of the embedding layer z, we define $\bar{z} := \sum_{i=1}^{N} z_i / N$ the average representation over the N samples and compute the covariance as follows:

$$C := \frac{1}{N} \sum_{i=1}^{N} (z_i - \bar{z})(z_i - \bar{z})^\top : \tag{6}$$

To plot the spectrum of the covariance matrix, we take the singularalue decomposition of the matrix ($C = USV^\top$) with S the diagonal of the singular values, which we plot in sorted order and logarithm scale in Figure 8.

This experiment demonstrates that the softmax normalization counters the dimensionality collapse that was discussed in Jing et al. (2022). Interestingly, the drop observed with SEM with 500 occurs at the index 2048 which is the dimensionality output of the ResNet-50 encoder.

Figure 8: Spectrum of the covariance matrix of the represention for BYOL and BYOL + SEM obtained with a ResNet-50 encoder.

D.4 Scaling the ResNet encoder for CIFAR-100

We perform a scaling experiment on CIFAR-100 where we compare the scaling behaviour of BYOL and BYOL + SEM. We evaluate the computational cost of the methods and the resulting downstream accuracy for a range of four resnets: ResNet-18, ResNet-50, ResNet-50 x2 and ResNet-50 x4. In Figure 9, we observe that SEM has a better scaling behaviour than the baseline, especially as we increase the width of the ResNet-50. For BYOL, we observe that the performance decays for

Figure 9: Scaling the ResNet encoder for CIFAR-100.

Table 20: Downstream accuracy of training a classi er with SEM normalization of the representation while using unormalized representation during pretraining. Experiments performed with a ResNet-50 encoder.

Pre-train model	Probe location	SEM ($\tau = 0.1$)	Accuracy
BYOL + Embed	Embedder	No	69.8
BYOL + Embed	Embedder	Yes	72.3
BYOL + SEM	Embedder	Yes	77.3

ResNet-50 with width x2 and x4. This is not unprecendented, as prior works as demonstrated other methods where scaling up the capacity of a model led to decrease in performance. When comparing the discrepancy with Figure 1, we attribute that to the fact that CIFAR-100 is a small dataset. In fact, we observe that the training accuracy stays constant to about 79% for all the ResNet-50 scales demonstrating over tting for the baseline BYOL. Nevertheless, SEM prevents the decrease in performance and even lead to further improved performance as we increase the scale of the ResNet-50.

D.5 The role of pretraining with SEM

We probe the downstream accuracy obtained of a model pre-trained without SEM and add SEM normalization only for the downstream classi cation. For this experiment, we take a pre-trained model with embedder (i.e. BYOL + embed) with $L = 5000$ and $V = 13$ and add the softmax normalization only for the downstream classi cation. We do not use SEM during pre-training. We observe that using SEM for downstream classi cation leads to an improvement even when the model is not pre-trained with SEM, demonstrating the utility of SEM downstream classi cation. However, we note that the performance of the model pre-trained without SEM is much weaker and thus demonstrates the imprtance of also pre-training using SEM.

E CIFAR-10 results

We con rm that our method also yield improvement on simpler datasets such as CIFAR-10. Here, we compare BYOL and BYOL + SEM on a ResNet-50 and observe and improvement of 1.6%.

Table 21: Downstream accuracy of training a classi er with SEM normalization of the representation while using unormalized representation during pretraining. Experiments performed with a ResNet-50 encoder.

Pre-train model	TOP-1 Accuracy
BYOL	94.2
BYOL + SEM	95.8

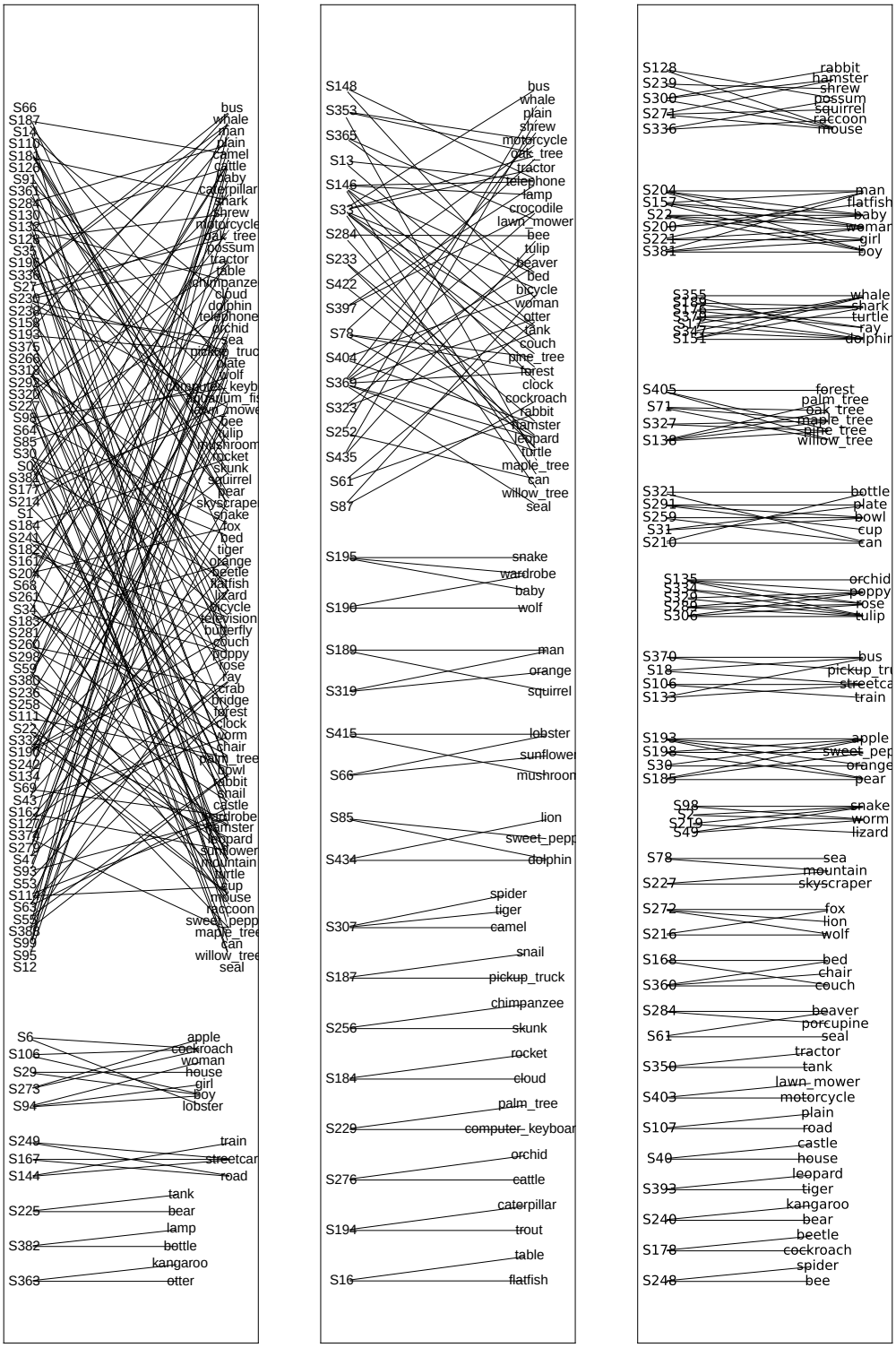
F CIFAR100SUPERCLASS

The 100 classes of CIFAR-100 (Krizhevsky, 2009) are grouped into 20 superclasses. The list of superclass for each class in Table 22

Table 22: Set of classes for each superclass on CIFAR-100.

Superclass	Classes
aquatic mammals	beaver, dolphin, otter, seal, whale
sh	aquarium sh, at sh, ray, shark, trout
owers	orchids, poppies, roses, sun owers, tulips
food containers	bottles, bowls, cans, cups, plates
fruit and vegetables	apples, mushrooms, oranges, pears, sweet peppers
household electrical devices	clock, computer keyboard, lamp, telephone, television
household furniture	bed, chair, couch, table, wardrobe
insects	bee, beetle, butter y, caterpillar, cockroach
large carnivores	bear, leopard, lion, tiger, wolf
large man-made outdoor things	bridge, castle, house, road, skyscraper
large natural outdoor scenes	cloud, forest, mountain, plain, sea
large omnivores and herbivores	camel, cattle, chimpanzee, elephant, kangaroo
medium-sized mammals	fox, porcupine, possum, raccoon, skunk
non-insect invertebrates	crab, lobster, snail, spider, worm
people	baby, boy, girl, man, woman
reptiles	crocodile, dinosaur, lizard, snake, turtle
small mammals	hamster, mouse, rabbit, shrew, squirrel
trees	maple, oak, palm, pine, willow
vehicles 1	bicycle, bus, motorcycle, pickup truck, train
vehicles 2	lawn-mower, rocket, streetcar, tank, tractor

G ADDITIONAL CIFAR-100 COHERENCE GRAPHS



(a) BYOL baseline (b) BYOL baseline with a large representation (c) BYOL + SEM

Figure 10: Comparison of the full semantic coherence graph W_5 between BYOL and BYOL + SEM.