Simplicial Embeddings in Self-Supervised Learning and Downstream Classification

Samuel Lavoie\textsuperscript{\diamond,†}, Christos Tsirigotis\textsuperscript{\diamond,†}, Max Schwarzer\textsuperscript{\diamond,†}, Ankit Vani\textsuperscript{\diamond,†}, Michael Noukhovitch\textsuperscript{\diamond,†}, Kenji Kawaguchi\textsuperscript{‡}, Aaron Courville\textsuperscript{\diamond,†,♣}
\textsuperscript{\diamond}Mila, †Université de Montréal, ‡National University of Singapore, ♣CIFAR Fellow
{samuel.lavoie.m,mnoukhov,aaron.courville}@gmail.com
{christos.tsirigotis,max.schwarzer,ankit.vani}@umontreal.ca
kenji@comp.nus.edu.sg

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 pretraining and imparts an inductive bias for group sparsity. For downstream classification, we formally prove that the SEM representation leads to better generalization than an 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

Over-complete representations are representations of an input that are non-unique combinations of a number of basis vectors greater than the input’s dimensionality [Lewicki and Sejnowski, 2000]. Mostly studied in the context of the sparse-coding literature [Gregor and LeCun, 2010; Goodfellow et al., 2012; Olshausen, 2013], sparse over-complete representations have been shown to increase stability in the presence of noise [Donoho et al., 2006], have applications in neuroscience [Olshausen and Field, 1996; Lee et al., 2007] and lead to more interpretable representations [Murphy et al., 2012; Fyshe et al., 2015; Faruqui et al., 2015]. However, the choice of basis vectors is generally assumed to be learned using traditional methods such as ICA [Teh et al., 2003] or fitting linear models [Lewicki and Sejnowski, 2000], limiting the expressive power of the encoding function.

Meanwhile, self-supervised learning (SSL) is an emerging family of methods that aims to learn an encoding of the data without manual supervision, such as through class labels, and using neural network encoders. Recent work [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 this demonstrates SSL’s efficacy, we demonstrate that sparse and overcomplete representations can further improve the downstream performance of these methods.

Inspired by work on language emergence, Dessì et al. [2021] propose to induce a discrete representation at the output of the encoder in a SSL model. Contrary to their work, we demonstrate that hard-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 vectors [Correia et al., 2019; Goyal et al., 2022], also alleviating the need to use high-variance estimators to back-propagate the gradient through the encoder. We refer to this embedding as Simplicial Embeddings (SEM) because the
softmax functions map the unnormalized representations onto $L$ simplices. The procedure to induce SEM is simple, efficient, and generally applicable.

The SSL pre-training phase, used with SEM, learns the set of $L$ approximately-sparse 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 and Ng, 2011], we find that the optimal sparsity for pre-training need not correspond to the optimal 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 leverage the SEM to control the representation’s expressivity via the softmax’s temperature.

Empirically, we provide evidences that SEM is applicable to most recent SSL methods and lead to a better representation. For the seven SSL methods probed [Chen et al., 2020; He et al., 2020; Grill et al., 2020; Caron et al., 2020, 2021; Zbontar et al., 2021; Bardes et al., 2022], SEM increases the accuracy by 2% to 4% on CIFAR-100 over the baseline without SEM. We observe constant improvement as we increase the number of vectors $L$ showing benefits of overcomplete representation in SEM. We also observe important improvement when training a SSL method with SEM on ImageNet on in-distribution test sets as well as several out-of-distribution test sets and transfer learning benchmarks, demonstrating the potential of SEM for large scale applications. Finally, we perform a qualitative analysis, and find that SEM learns features that are closely aligned to the semantic categories extant in the data, demonstrating evidence of disentangled and more interpretable representations, as it was previously observed in complete representations [Faruqui et al., 2015].

2 Simplicial Embeddings

Simplicial Embeddings (SEM) are representations that can be integrated easily into a contrastive learning model [Hjelm et al., 2019; Chen et al., 2020], 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 SEM after the encoder and before the projector and the rest is unchanged as shown in Figure 2c.

To produce an SEM representation, the encoder’s output $e_{θ}$ is embedded into $L$ vectors $z_i \in \mathbb{R}^V$. A temperature parameter $τ$ scales $z_i$ then a softmax re-normalizes each vector $z_i$ to produce $\bar{z}_i$. Finally, the normalized vectors $\bar{z}_i$ are concatenated to produce the vector $\hat{z}$ of length $L \cdot V$. Formally, the re-normalization is as follows:

$$\bar{z}_i := σ_τ(z_i), \quad σ_τ(z_i) = \frac{e^{z_i/τ}}{\sum_{k=1}^{V} e^{z_k/τ}}, \quad \hat{z} := \text{Concat}(\bar{z}_1, \ldots, \bar{z}_L), \quad \forall i \in [L], \forall j \in [V]. \quad (1)$$

3 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]. On IMAGENET [Deng et al., 2009], we study the effect of SEM on robustness and transfer learning datasets. Finally, we present evidences that features that result from SEMs appear to be more naturally aligned to the semantic categories found in the data.

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. In our experiments, the embedder is a linear layer followed by BatchNorm [Ioffe and 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 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 use a batch size of 256, and train models for 200 epochs on IMAGENET and 1000 epochs on CIFAR-100.
3.1 SEM improves on downstream classification

We evaluate the effect of adding SEMs in seven modern SSL approaches. We take standard SimCLR [Chen et al., 2020], 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 in Table 1. The reported numbers are the means and the standard deviations over 5 random seeds. For every SSL method, using SEMs improves the baseline methods by a margin of 2% to 4% demonstrating that SEM is a general approach that improves in-distribution generalization for SSL methods.

<table>
<thead>
<tr>
<th>Model</th>
<th>Baseline</th>
<th>SEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimCLR</td>
<td>65.8 ± 0.3</td>
<td>69.5 ± 0.2</td>
</tr>
<tr>
<td>MoCo</td>
<td>69.3 ± 0.3</td>
<td>71.0 ± 0.3</td>
</tr>
<tr>
<td>BYOL</td>
<td>70.7 ± 0.2</td>
<td>73.9 ± 0.2</td>
</tr>
<tr>
<td>Barlow-Twins</td>
<td>70.7 ± 0.3</td>
<td>73.0 ± 0.2</td>
</tr>
<tr>
<td>SwAV</td>
<td>64.6 ± 0.3</td>
<td>67.7 ± 0.2</td>
</tr>
<tr>
<td>DINO</td>
<td>66.8 ± 0.3</td>
<td>69.2 ± 0.3</td>
</tr>
<tr>
<td>VicReg</td>
<td>68.5 ± 0.2</td>
<td>71.4 ± 0.4</td>
</tr>
</tbody>
</table>

Table 1: Linear probe accuracy on CIFAR-100 trained for 1000 epochs with ResNet-18 encoder. We compare the test accuracy of several SSL models with and without SEM. The baseline models are taken from [da Costa et al., 2021]. The SEM normalized output ($z_0$) is used for the linear probe with SEM. **Boldface** indicates highest accuracy. Green rows indicate a SSL method + SEM.

3.2 SEM improvement on large-scale datasets with ImageNet

We demonstrate that SEM improves the accuracy on large scale datasets, such as **IMAGE**. We demonstrate that SEM generally improves the test set accuracy on several robustness test sets, transfer learning datasets and semi-supervised via fine-tuning with 1% and 10% of the data. The embedding is pre-trained for 200 epochs using the BYOL SSL procedure.

<table>
<thead>
<tr>
<th>Model</th>
<th>IN</th>
<th>IN-V2</th>
<th>IN-R</th>
<th>IN-C</th>
<th>FLOPs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ResNet50:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BYOL</td>
<td>70.6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>4.1e9</td>
</tr>
<tr>
<td>BYOL*</td>
<td>71.9</td>
<td>59.2</td>
<td>18.8</td>
<td>39.5</td>
<td>4.1e9</td>
</tr>
<tr>
<td><strong>BYOL+SEM:</strong></td>
<td>74.1</td>
<td>61.2</td>
<td>22.1</td>
<td>43.4</td>
<td>4.7e9</td>
</tr>
<tr>
<td><strong>ResNet50-x2:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BYOL</td>
<td>74.2</td>
<td>62.1</td>
<td>22.2</td>
<td>47.3</td>
<td>1.1e10</td>
</tr>
<tr>
<td>BYOL+SEM</td>
<td>75.9</td>
<td>63.7</td>
<td>23.5</td>
<td>48.8</td>
<td>1.2e10</td>
</tr>
<tr>
<td><strong>ResNet50-x4:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BYOL</td>
<td>75.8</td>
<td>64.0</td>
<td>22.9</td>
<td>49.8</td>
<td>3.7e10</td>
</tr>
<tr>
<td>BYOL+SEM</td>
<td>77.2</td>
<td>64.8</td>
<td>25.1</td>
<td>52.0</td>
<td>3.8e10</td>
</tr>
</tbody>
</table>

Table 2: Test accuracies of a linear probe trained with the **IMAGE** samples on a pre-trained representation for 200 epochs. * Taken from [Chen and He, 2020]

Table 3: Transfer learning accuracy by training a linear probe on a pre-trained representation with **IMAGE** for 200 epochs.

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-1</th>
<th>Top-5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BYOL</strong></td>
<td>61.6</td>
<td>67.5</td>
</tr>
<tr>
<td><strong>BYOL+SEM</strong></td>
<td>56.7</td>
<td>69.9</td>
</tr>
</tbody>
</table>

Table 4: Semi-supervised learning top-1 and top-5 accuracy by fine-tuning a model on ImageNet.

3.3 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 $W \in \mathbb{R}^{N \times C}$, 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 $W_K$.

The qualitative analysis is given by plotting the subset $W_K$, obtained by taking the top 5 features for each class. We present a subset of the graph for BYOL+SEM in Figure 1a and for BYOL in Figure 1b. 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
Figure 1: Semantic coherence of the features. (a) and (b) Subset of $W_K$, the bipartite graph of the most important features shared between at least two classes of a classifier trained on BYOL + SEM features in (a) and BYOL on the encoded features in (b). The connected components emerge without additional interventions in BYOL + SEM. (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.

First set of connected components are flowers, 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 on $W_K$ 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, define the operation $\text{is\_super}(c_i, c_j)$ 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 7 in the Appendix. We measure semantic coherence as follows:

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

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 on 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 and Oja, 2000] in an attempt to disentangle the features. In Figure 6, we plot the full graph $W_K$ 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.

4 Conclusion

SEMs are representations that can be obtained by embedding partitions of a latent representation using a softmax operation. This simple modification leads to improved generalization on downstream classification for several SSL methods. Furthermore, our semantic coherence analysis indicates that SEMs can naturally disentangle the semantic categories of the data without explicit training objectives. We hope that this work motivates the use of SEM with pre-training to learn discrete representation useful for downstream applications and the study of architectural inductive biases for SSL representations towards more explainable and performant models.
Acknowledgments and Disclosure of Funding

The authors are grateful for the insightful discussions with Xavier Bouthillier, Hattie Zhou, Sébastien Lachapelle, Tristan Deleu, Yuchen Lu, Eeshan Dhekane, and Devon Hjelm. 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


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 $\sigma$, the softmax operation with temperature $\tau$. The normalized vectors are concatenated into the vector $\hat{z}$. (b) Normalized histogram of the entropies $H(\bar{z}_i)$ of each simplex $\bar{z}_i$ for the sample in CIFAR’s training dataset at the end of pre-training with various $\tau$. 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.

A Properties of the Simplicial Embeddings

In this subsection, we discuss some of the properties of the Simplicial Embeddings during pre-training and for the downstream tasks.

A.1 Inductive bias towards sparsity during pre-training

In SEM, $L$ controls the numbers of vectors and $V$ controls the number of component that each vectors may have. As such, the higher $V$ is, the sparser the representation is and also the stronger the bias toward sparsity as discussed in Vaswani et al. [2017]; Wang et al. [2021]. During pre-training, the constraints of the simplex biases each vector towards sparsity by creating a zero-sum competition between the elements. In order for an element of a vector to increase by $\alpha$, then the other elements must decrease by $\alpha$ and all elements are bounded by 0. For networks to learn useful features, they must prioritize some at the expense of others. For SSL methods with a target network, the temperature for the target network can be different of the online network’s as no gradient is back-propagated.

To visualize the effect of the temperature on SEM after pre-training, we interpret each simplex as a probability mass function $p(\bar{z}_i)$ where, for all $i \in [L]$, $\sum_{j=1}^V p(\bar{z}_{ij}) = 1$ and $p(\bar{z}_{ij}) \geq 0 \forall j$. The entropy of a simplex $\bar{z}_i$, defined $H(\bar{z}_i) := -\sum_{j=1}^V p(\bar{z}_{ij}) \log p(\bar{z}_{ij})$, informs whether the simplex is a sparse or a dense vector. That is, if $H(\bar{z}_i^{(x)}) = 0$ then the vector is one-hot. On the other hand, if $H(\bar{z}_i^{(x)}) = \ln(V)$ then the vector is dense and uniform. While the temperature $\tau_p$ 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 $H(\bar{z}_i)$, for a given $\tau_p$, 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 $\tau_p$ for optimal inductive bias: $\tau_p$ too small yields vanishing
We also make the following assumptions. We assume that there exists

\[ g \]

we propose to partition the matrix multiplication into \( n \) small non-overlapping matrix multiplications. Formally, let \( v \in \mathbb{R}^{b \times m_o}, w \in \mathbb{R}^{m_o \times o} \) and \( y = v \cdot w \) be the fully connected matrix multiplication. Instead, we partition \( v \) into \( n \) blocks with \( v^i \in \mathbb{R}^{b \times \frac{m_o}{n}} \) and define \( n \) smaller \( w^i \in \mathbb{R}^{\frac{m_o}{n} \times o} \), where \( i \in [L] \) is the \( i^{th} \) block. Then, we perform a batch matrix multiplication of \( v^i \) and \( w^i \) that we concatenate as follows: \( y^i = v^i \cdot w^i \) and \( \tilde{y}^i = \text{Concat}([y^1, \ldots, y^n]) \). Thus, the amount of parameters of this matrix multiplication scales in \( \mathcal{O}(\frac{m_o^2}{n}) \), allowing us to reduce the memory consumption by increasing \( n \), the number of blocks.

A.3 SEM improvement on the generalization of the downstream classifier

In this subsection, we mathematically analyze the effect of using SEM for downstream classification. We aim to understand the benefit of training a downstream classifier with SEM normalized input compared to a baseline classifier with unnormalized input. In summary, we show that: (1) there is a trade-off between the training loss and the generalization gap, which is controlled by the value of \( \tau_d \), (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 improve or stay constant as \( L \) and \( V \) increase. In the remaining of this subsection, we consider \( \tau = \tau_d \).

Notation We consider a training dataset \( S = \{ z^{(i)}, y^{(i)} \}_{i=1}^n \) of \( n \) samples that is used for supervised training of a classifier using the representation \( z \), which are extracted from the pre-trained model\(^1\) and the corresponding label \( y \). Let \( g \) represent the downstream classifier trained on the representation. We can define a baseline model where \( g \) is trained without normalization as \( f_{\text{base}}(z) = g(z) \) and the corresponding model trained with the SEM normalization of the representation’s features as \( f_{\text{SEM}}(z) = (g \circ \sigma_{\tau})(z) \). Here, \( \sigma_{\tau} \) is applied to each vector \( z_i \). To compare the quality of the base model and the model with SEM normalization, we analyze the generalization gap \( \mathbb{E}_{z,y}[l(f(z), y)] - \frac{1}{n} \sum_{i=1}^n l(f(z^{(i)}), y^{(i)}) \) for each \( f \in \{ f_{\text{SEM}}(\tau), f_{\text{base}} \} \), where \( l : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}_{\geq 0} \) is the per-sample loss and \( f_{\text{SEM}}(\tau), f_{\text{base}} \) are the models obtained from fitting the dataset \( S \).

To simplify the notation, we consider the normalization to \([-1, +1] \); i.e., the encoder’s output \( z \in \mathcal{Z} = [-1, +1]^{L \times V} \). Next, we define \( Q_i = \{ q \in [-1, +1]^V : i = \arg \max_{j \in [V]} q_j \} \), the partition of the space \([-1, +1]^V \). I.e. we have \( V \) partitions \( Q_i \) with \( i \in [V] \) and \( Q_i \) is the partition of the space where \( i = \arg \max_{j \in [V]} q \). We use \( Q_i \) to define the following measure that allows us to understand the effect of the SEM normalization on the representation given to the classifier \( g \) and thus to compare \( f_{\text{SEM}} \) and \( f_{\text{base}} \), a model with and without SEM normalization respectively:

\[
\varphi(\sigma_f) = \sup_{i \in [V]} \sup_{q,q' \in Q_i} \| \sigma_f(q) - \sigma_f(q') \|_2
\]

where \( \sigma_{\text{f}_{\text{SEM}}} = \sigma_{\tau} \) and \( \sigma_{\text{f}_{\text{base}}} \) is the identity function. Here, \( \sigma_{\tau}(q) = \frac{\rho_i(q)}{\sum_{j=1}^{V} \rho_j(q)} \) for \( j = 1, \ldots, V \). Intuitively, \( \varphi \) is a measure on the expressivity of the representation and depends on \( V \) and \( \tau \).

We also make the following assumptions. We assume that there exists \( \Delta > 0 \) such that for any \( i \in [L] \), if \( k = \arg \max_{j \in [V]} z_{ij} \), then \( z_{ik} \geq z_{ij} + \Delta \) for any \( j \neq k \). Since \( \Delta \) can be arbitrarily small (e.g., much smaller than machine precision), this assumption typically holds in practice. Next, we define \( B \) to be the upper bound on the per-sample loss such that \( l(f(z), y) \leq B \) for all \( f \in \mathcal{H} \) and for all \( z, y \in \mathcal{Z} \times \mathcal{Y} \), where \( \mathcal{H} \) is the union of the hypothesis spaces of \( f_{\text{SEM}}(\tau) \) and \( f_{\text{base}} \). For example, \( B = 1 \) for the 0-1 loss.

Finally, we define \( \mathcal{G}_S \) to be the set of classifiers \( g \) returned by the training algorithm using dataset \( S \), and \( R \) to be the Lipschitz constant of \( l_y \circ g \) for all \( y \in \mathcal{Y} \) and \( g \in \mathcal{G}_S \); i.e., \( |l_y \circ g(\sigma_f(z)) - (l_y \circ g(\sigma_{\tau}))(z)| \leq R \).
We first decompose the generalization gap into two terms using the following lemma:

\[ g(y) = g(f(z')) \leq R[\|g(f) - f(z')\|_F, \text{ where } l_y(g \circ f(z)) = l(g \circ f(z), y). \] We denote by \( c > 0 \) a constant in \((n, f, \mathcal{H}, \delta, \mathcal{H}, \tau, S)\). 

Using the established notation, Theorem 1 illuminates the advantage of SEM and the effect of the hyper-parameter \( \tau \) on the performance of the downstream classifier. We present the proof in Appendix B and we present empirical evidence of the theorem’s prediction in Figure S.

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

\[
E_{z,y}[l(f_S(z), y)] \leq \frac{1}{n} \sum_{i=1}^{n} l(f_S(z^{(i)}), y^{(i)}) + R\sqrt{L\varphi(f_S)} + c\sqrt{\ln(2/\delta)/n}.
\]

Moreover,

\[
\varphi(f_{\text{SEM}(\tau)}) \rightarrow 0 \text{ as } \tau \rightarrow 0 \quad \text{and} \quad \varphi(f_{\text{SEM}(\tau)}) - \varphi(f_{\text{base}}) \leq \frac{3}{4}(1 - V) < 0 \quad \forall \tau > 0.
\]

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 \( c\sqrt{\ln(2/\delta)/n} \). Since \( c \) is a constant in \((n, f, \mathcal{H}, \delta, \mathcal{H}, \tau, S)\), 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., \( \varphi(f_{\text{SEM}(\tau)}) \rightarrow 0 \) as \( \tau \rightarrow 0 \). Also, for any \( \tau > 0 \), the second term with SEM is strictly smaller than that without SEM as \( \varphi(f_{\text{SEM}(\tau)}) - \varphi(f_{\text{base}}) \leq \frac{3}{4}(1 - V) < 0 \) and demonstrates that the improvement due to SEM is expected to asymptotically increase as \( V \) increases. Also, \( L \) is a multiplicative constant of \( \varphi \) which shows that as \( L \) increases, the expect 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 \( \tau \). When \( \tau \to 0 \), the second term goes to zero, but the training loss (the first term) can increase due to the reduction in expressivity and increased difficulty in optimization. Thus, \( \tau \) should be chosen to optimally balance this trade-off.

**B Proof of Theorem 1**

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

\[
\tilde{C}_{y,k_1,\ldots,k_L} = \{(z, \bar{y}) \in \mathcal{Z} \times \mathcal{Y} : \bar{y} = y, k_j = \arg \max_{t \in [V]} z_j,t \quad \forall j \in [L]\}
\]

and

\[
\hat{Z}_{k_1,\ldots,k_L} = \{z \in \mathcal{Z} : k_j = \arg \max_{t \in [V]} z_j,t \quad \forall j \in [L]\}.
\]

We then define \( C_k \) to be the flatten version of \( \tilde{C}_{y,k_1,\ldots,k_L} \); i.e., \( \{C_k\}_{k=1}^{L} = \{\tilde{C}_{y,k_1,\ldots,k_L} \}_{y \in \mathcal{Y}, k_1,\ldots,k_L \in [V]} \) with \( C_1 = \tilde{C}_{1,\ldots,1} \), \( C_2 = \tilde{C}_{2,\ldots,2} \), \( C_3 = \tilde{C}_{3,\ldots,3} \), \( C_4 = \tilde{C}_{4,\ldots,4} \), and so on. Similarly, define \( \hat{Z}_k \) to be the flatten version of \( \hat{Z}_{k_1,\ldots,k_L} \). We also use \( Q_k \) as \( Q_k = \{-1,1\}^V \), \( I_k := I_k = \{i \in [n] : r_i \in C_k\} \), and \( \alpha_k(h) := E_r[l(h, r)] \). Moreover, we define \( \varphi(f_{\text{base}}) = \sup_{i \in [V]} \sup_{q,q' \in Q_i} \|\sigma(\bar{q}) - \sigma(\bar{q}')\|_{2}^{2} \)

and

\[
\varphi(f_{\text{SEM}(\tau)}) = \sup_{i \in [V]} \sup_{q,q' \in Q_i} \|\sigma(\bar{q}) - \sigma(\bar{q}')\|_{2}^{2} \text{ where } \sigma(\bar{q}) = \frac{\epsilon_q}{\epsilon_{q'}^{0/\tau}} \sum_{t=1}^{\epsilon_q} \epsilon_t^{0/\tau}
\]

for \( j = 1,\ldots,V \).

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

**Lemma 1.** For any \( \delta > 0 \), with probability at least \( 1 - \delta \), the following holds for all \( h \in \mathcal{H} \):

\[
E_{r}[\ell(h, r)] - \frac{1}{n} \sum_{i=1}^{n} \ell(h, r_i) \leq \frac{1}{n} \sum_{k=1}^{K} |I_k| \left( \alpha_k(h) - \frac{1}{|I_k|} \sum_{i \in I_k} \ell(h, r_i) \right) + c\sqrt{\ln(2/\delta)/n}.
\]
Proof. We first write the expected error as the sum of the conditional expected error:

\[ \mathbb{E}_r[\ell(h, r)] = \sum_{k=1}^{K} \mathbb{E}_r[\ell(h, r)|r \in C_k] \Pr(r \in C_k) = \sum_{k=1}^{K} \mathbb{E}_{r_k}[\ell(h, r_k)] \Pr(r \in C_k), \]

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:

\[ \mathbb{E}_r[\ell(h, r)] - \frac{1}{n} \sum_{i=1}^{n} \ell(h, r_i) \] (4)

The second term in the right-hand side of (4) 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

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

Substituting these into equation (4) yields

\[ \mathbb{E}_r[\ell(h, r)] - \frac{1}{n} \sum_{i=1}^{n} \ell(h, r_i) \] (5)

\[ = \sum_{k=1}^{K} \mathbb{E}_{r_k}[\ell(h, r_k)] \left( \Pr(r \in C_k) - \frac{|I_k|}{n} \right) + \frac{1}{n} \sum_{k=1}^{K} |I_k| \left( \mathbb{E}_{r_k}[\ell(h, r_k)] - \frac{1}{|I_k|} \sum_{i \in I_k} \ell(h, r_i) \right) \]

\[ \leq B \sum_{k=1}^{K} \left| \Pr(r \in C_k) - \frac{|I_k|}{n} \right| + \frac{1}{n} \sum_{k=1}^{K} |I_k| \left( \mathbb{E}_{r_k}[\ell(h, r_k)] - \frac{1}{|I_k|} \sum_{i \in I_k} \ell(h, r_i) \right) \]

By using the Bretagnolle-Huber-Carl inequality \cite[van der Vaart and Wellner 1996 A6.6 Proposition]{1996}, we have that for any \( \delta > 0 \), with probability at least \( 1 - \delta \),

\[ \sum_{k=1}^{K} |I_k| \left| \Pr(r \in C_k) - \frac{|I_k|}{n} \right| \leq \sqrt{2K \ln(2/\delta) \over n}. \]

(6)

Here, notice that the term of \( \sum_{k=1}^{K} \Pr(r \in C_k) - \frac{|I_k|}{n} \) does not depend on \( h \in \mathcal{H} \). Moreover, note that for any \( (f, h, M) \) such that \( M > 0 \) and \( B \geq 0 \) for all \( X \), we have that \( \mathbb{P}(f(X) \geq M) \geq \mathbb{P}(f(X) > M) \geq \mathbb{P}(B f(X) + h(X) > BM + h(X)) \), where the probability is with respect to the randomness of \( X \). Thus, by combining (5) and (6), we have that for any \( h \in \mathcal{H} \), for any \( \delta > 0 \), with probability at least \( 1 - \delta \), the following holds for all \( h \in \mathcal{H} \),

\[ \mathbb{E}_r[\ell(h, r)] - \frac{1}{n} \sum_{i=1}^{n} \ell(h, r_i) \leq \frac{1}{n} \sum_{k=1}^{K} |I_k| \left( \alpha_k(h) - \frac{1}{|I_k|} \sum_{i \in I_k} \ell(h, r_i) \right) + c \sqrt{\ln(2/\delta) \over n}. \]

\[ \square \]

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

Lemma 2. For any \( f \in \left\{ f_{SEM(r)}^{S}, f_{base}^{S} \right\} \),

\[ \frac{1}{n} \sum_{k=1}^{K} |I_k| \left( \alpha_k(f) - \frac{1}{|I_k|} \sum_{i \in I_k} \ell(f, r_i) \right) \leq R \sqrt{L \varphi(f)}. \]
Therefore, for any $f, g \in \mathcal{G}_S$, by using the Lipschitz continuity, boundedness, and non-negativity,

$$
\sup_{r, r' \in C_k} |\ell(f, r) - \ell(f, r')| = \sup_{y \in \mathcal{Y}} \sup_{z, z' \in \mathcal{Z}_k} |(l_y \circ g_{\mathcal{G}_S}(\tau))(z) - (l_y \circ g_{\mathcal{G}_S}(\tau))(z')| \\
\leq R \sup_{z, z' \in \mathcal{Z}_k} \|\sigma_{\tau}(z) - \sigma_{\tau}(z')\|_F \\
= R \sup_{z, z' \in \mathcal{Z}_k} \sqrt{\sum_{i=1}^{L} \sum_{j=1}^{V} (\sigma_{\tau}(z_{i, j}) - \sigma_{\tau}(z'_{i, j}))^2} \\
\leq R \sqrt{\sum_{i=1}^{L} \sum_{j=1}^{V} \|\sigma_{\tau}(q) - \sigma_{\tau}(q')\|_2^2} \\
= R \sqrt{L\varphi(f_{\mathcal{G}_S}(\tau))}.
$$

Similarly, if $f = f^S_{\mathcal{G}_S}$, since $g_{\mathcal{G}_S} \in \mathcal{G}_S$, by using the Lipschitz continuity, boundedness, and non-negativity,

$$
\sup_{r, r' \in C_k} |\ell(f, r) - \ell(f, r')| = \sup_{y \in \mathcal{Y}} \sup_{z, z' \in \mathcal{Z}_k} |(l_y \circ g_{\mathcal{G}_S}(\tau))(z) - (l_y \circ g_{\mathcal{G}_S}(\tau))(z')| \\
\leq R \sup_{z, z' \in \mathcal{Z}_k} \|z - z'\|_F \\
\leq R \sqrt{L\varphi(f_{\mathcal{G}_S})}.
$$

Therefore, for any $f \in \{f^S_{\mathcal{G}_S}, f^S_{\mathcal{G}_S}\}$,\n
$$
\frac{1}{n} \sum_{k=1}^{K} \left| I_k \right| \left( \alpha_k(f) - \frac{1}{\left| I_k \right|} \sum_{i \in I_k} \ell(f, r_i) \right) \leq \frac{1}{n} \sum_{k=1}^{K} \left| I_k \right| R \sqrt{L\varphi(f)} = R \sqrt{L\varphi(f)}.
$$

Combining Lemma[4] 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 $f \in \{f^S_{\mathcal{G}_S}, f^S_{\mathcal{G}_S}\}$:

$$
\mathbb{E}_r[\ell(f, r)] - \frac{1}{n} \sum_{i=1}^{n} \ell(f, r_i) \leq R \sqrt{L\varphi(f)} + \sqrt{\frac{\ln(2/\delta)}{n}}.
$$
Proof. This follows directly from combining Lemma 1 and Lemma 2.

We now provide an upper bound on $\varphi(f_{\text{SEM}(\tau)}^S)$ in the following lemma:

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

$$\varphi(f_{\text{SEM}(\tau)}^S) \leq \left| \frac{1}{1 + (V - 1)e^{-2\tau}} - \frac{1}{1 + (V - 1)e^{-\Delta/\tau}} \right|^2 + (V - 1)\left| \frac{1}{1 + e^{\Delta/\tau}(1 + (V - 2)e^{-2\tau}/\tau)} - \frac{1}{1 + e^{\Delta/\tau}(1 + (V - 2)e^{-\Delta/\tau}/\tau)} \right|^2.$$ 

Proof. Recall the definition:

$$\varphi(f_{\text{SEM}(\tau)}^S) = \sup_{i \in [V]} \sup_{q, q' \in Q_i} \|\sigma_\tau(q) - \sigma_\tau(q')\|_2^2,$$

where

$$\sigma_\tau(q)_j = \frac{e^{q_j/\tau}}{\sum_{i=1}^V e^{q_i/\tau}},$$

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

$$\varphi(f_{\text{SEM}(\tau)}^S) = \sup_{q, q' \in Q_1} \|\sigma_\tau(q) - \sigma_\tau(q')\|_2^2.$$

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 $\delta_i, \delta'_i > 0$ such that

$$q_i = q_1 - \delta_i$$

and

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

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

$$\delta_i, \delta'_i \geq \Delta > 0.$$

Thus, we can rewrite

$$\sum_{i=1}^V e^{q_i/\tau} = e^{q_1/\tau} + \sum_{i=2}^V e^{(q_1 - \delta_i)/\tau}$$

$$= e^{q_1/\tau} + e^{q_1/\tau} \sum_{i=2}^V e^{-\delta_i/\tau}$$

$$= e^{q_1/\tau} \left( 1 + \sum_{i=2}^V e^{-\delta_i/\tau} \right)$$

Similarly,

$$\sum_{i=1}^V e^{q'_i/\tau} = e^{q'_1/\tau} \left( 1 + \sum_{i=2}^V e^{-\delta'_i/\tau} \right).$$

Using these,

$$\sigma_\tau(q)_1 = \frac{e^{q_1/\tau}}{\sum_{i=1}^V e^{q_i/\tau}} = \frac{e^{q_1/\tau}}{e^{q_1/\tau} \left( 1 + \sum_{i=2}^V e^{-\delta_i/\tau} \right)} = \frac{1}{1 + \sum_{i=2}^V e^{-\delta_i/\tau}}$$
and for all $j \in \{2, \ldots, V\}$,

$$
\sigma_\tau(q_j) = \frac{e^{\eta_j/\tau}}{\sum_{t=1}^{V} e^{\eta_t/\tau}}
= \frac{e^{\eta_j/\tau} (1 + \sum_{i=2}^{V} e^{-\delta_i/\tau})}{1 + \sum_{i=2}^{V} e^{-\delta_i/\tau}}
= \frac{1}{1 + \sum_{i=2}^{V} e^{-\delta_i/\tau}}
\sum_{t=1}^{V} e^{\eta_t/\tau}
\]$$

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

$$
\sigma_\tau(q_j') = \frac{1}{1 + \sum_{i=2}^{V} e^{-\delta_i'/\tau}},
$$

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

$$
\sigma_\tau(q_j') = \frac{1}{1 + \sum_{i=2}^{V} e^{-\delta_i'/\tau}}.
$$

Using these, for any $q, q' \in Q_1$,

$$
|\sigma_\tau(q) - \sigma_\tau(q')|_1 = \left| \frac{1}{1 + \sum_{i=2}^{V} e^{-\delta_i/\tau}} - \frac{1}{1 + \sum_{i=2}^{V} e^{-\delta_i'/\tau}} \right|
\leq \left| \frac{1}{1 + \sum_{i=2}^{V} e^{-2/\tau}} - \frac{1}{1 + \sum_{i=2}^{V} e^{-\Delta/\tau}} \right|
= \left| \frac{1}{1 + (V-1)e^{-2/\tau}} - \frac{1}{1 + (V-1)e^{-\Delta/\tau}} \right|
$$

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

$$
|\sigma_\tau(q_j) - \sigma_\tau(q_j')|_1 = \left| \frac{1}{1 + e^{\delta_j/\tau} + \sum_{t \in I_j} e^{(\delta_j - \delta_t)/\tau}} - \frac{1}{1 + e^{\delta_j'/\tau} + \sum_{t \in I_j} e^{(\delta_j - \delta_t)/\tau}} \right|
\leq \left| \frac{1}{1 + e^{\Delta/\tau} + \sum_{t \in I_j} e^{(\Delta - 2)/\tau}} - \frac{1}{1 + e^{2/\tau} + \sum_{t \in I_j} e^{(2 - \Delta)/\tau}} \right|
= \left| \frac{1}{1 + e^{\Delta/\tau} + (V-2)e^{(\Delta - 2)/\tau}} - \frac{1}{1 + e^{2/\tau} + (V-2)e^{(2 - \Delta)/\tau}} \right|
= \left| \frac{1}{1 + e^{\Delta/\tau}(1 + (V-2)e^{-2/\tau})} - \frac{1}{1 + e^{2/\tau}(1 + (V-2)e^{-\Delta/\tau})} \right|.
$$

By combining these,

$$
\sup_{q,q' \in Q_1} \|\sigma_\tau(q) - \sigma_\tau(q')\|_2^2
= \sup_{q,q' \in Q_1} \sum_{j=1}^{V} |\sigma_\tau(q_j) - \sigma_\tau(q_j')|^2
\leq \left| \frac{1}{1 + (V-1)e^{-2/\tau}} - \frac{1}{1 + (V-1)e^{-\Delta/\tau}} \right|^2
+ (V-1) \left| \frac{1}{1 + e^{\Delta/\tau}(1 + (V-2)e^{-2/\tau})} - \frac{1}{1 + e^{2/\tau}(1 + (V-2)e^{-\Delta/\tau})} \right|^2.
$$
Using the previous lemma, we will conclude the asymptotic behavior of $\varphi(f_{SEM(\tau)}^S)$ in the following lemma:

**Lemma 5.** It holds that

$$\varphi(f_{SEM(\tau)}^S) \to 0 \text{ as } \tau \to 0.$$  

**Proof.** Using Lemma 4

$$\lim_{\tau \to 0} \varphi(f_{SEM(\tau)}^S) \leq \lim_{\tau \to 0} \left| \frac{1}{1 + (V - 1)e^{-2/\tau}} - \frac{1}{1 + (V - 1)e^{-\Delta/\tau}} \right|^2$$

$$+ n(V - 1) \lim_{\tau \to 0} \left| \frac{1}{1 + e^{\Delta/\tau}(1 + (V - 2)e^{-2/\tau})} - \frac{1}{1 + e^{2/\tau}(1 + (V - 2)e^{-\Delta/\tau})} \right|^2.$$

Moreover,

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

and

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

Therefore,

$$\lim_{\tau \to 0} \varphi(f_{SEM(\tau)}^S) \leq 0.$$

Since $\varphi(f_{SEM(\tau)}^S) \geq 0$, this implies the statement of this lemma.

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

**Lemma 6.** For any $\tau > 0$,

$$\varphi(f_{SEM(\tau)}^S) - \varphi(f_{base}^S) \leq \frac{3}{4}(1 - V) < 0.$$  

**Proof.** From Lemma 4 for any $\tau > 0$,

$$\varphi(f_{SEM(\tau)}^S) \leq \left| \frac{1}{1 + (V - 1)e^{-2/\tau}} - \frac{1}{1 + (V - 1)e^{-\Delta/\tau}} \right|^2$$

$$+ n(V - 1) \left| \frac{1}{1 + e^{\Delta/\tau}(1 + (V - 2)e^{-2/\tau})} - \frac{1}{1 + e^{2/\tau}(1 + (V - 2)e^{-\Delta/\tau})} \right|^2.$$

$$\leq \left| \frac{1}{1 + (V - 1)e^{-2/\tau}} - \frac{1}{1 + (V - 1)} \right|^2$$

$$+ (V - 1) \left| \frac{1}{1 + (V - 1)e^{-2/\tau}} - \frac{1}{1 + e^{2/\tau}(1 + (V - 2))} \right|^2$$

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

$$\leq \left| \frac{1}{V} - \frac{1}{2} \right|^2 + (V - 1) \left| \frac{1}{4} \right|^2.$$
Recall the definition of
\[ \varphi(f_{\text{base}}^S) = \sup_{i \in [V]} \sup_{q,q' \in Q_i} \| q - q' \|_2^2. \]

By choosing an element in the set over which the supremum is taken, for any \( \delta \geq \Delta > 0 \),
\[ \varphi(f_{\text{base}}^S) \geq \sup_{q,q' \in Q_1} \| q - q' \|_2^2 \geq \| \hat{q} - \hat{q}' \|_2^2 = \sum_{j=1}^{V} (\hat{q}_j - \hat{q}'_j)^2 = (2 - \delta)^2 V, \]
where \( \hat{q}_1 = 1, \hat{q}_j = 1 - \delta \) for \( j \in \{2, \ldots, V\} \), \( \hat{q}'_1 = \delta - 1 \), and \( \hat{q}'_j = -1 \) for \( j \in \{2, \ldots, V\} \).

By combining those, for for any \( \tau > 0 \) and \( \Delta \geq \Delta > 0 \),
\[ \varphi(f_{\text{SEM}(\tau)}^S) - \varphi(f_{\text{base}}^S) \leq \left( \frac{1}{1 - \frac{1}{V}} \right)^2 + (V - 1) \frac{1}{4} - (2 - \delta)^2 V \]
\[ \leq 1 + \frac{1}{4} V - \frac{1}{4} - (2 - \delta)^2 V \]
\[ = \frac{3}{4} + \frac{1}{4} V - (2 - \delta)^2 V \]
\[ = \frac{3}{4} - V \left( (2 - \delta)^2 - \frac{1}{4} \right) \]
\[ \leq \frac{3}{4} - V \left( 1 - \frac{1}{4} \right) \]
\[ = \frac{3}{4} (1 - V) \]

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

**Theorem 1.** Let \( V \geq 2 \). For any \( \delta > 0 \), with probability at least \( 1 - \delta \), the following holds for any \( f_S \in \{ f_{\text{SEM}(\tau)}^S, f_{\text{base}}^S \} \):
\[ \mathbb{E}_{z,y}[l(f_S(z), y)] \leq \frac{1}{n} \sum_{i=1}^{n} l(f_S(z^{(i)}), y^{(i)}) + R \sqrt{L \varphi(f_S) + c \frac{\ln(2/\delta)}{n}}. \]

Moreover,
\[ \varphi(f_{\text{SEM}(\tau)}^S) \to 0 \ as \ \tau \to 0 \ and \ \varphi(f_{\text{SEM}(\tau)}^S) - \varphi(f_{\text{base}}^S) \leq \frac{3}{4} (1 - V) < 0 \ \forall \tau > 0. \]

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

**C Additional experiments on CIFAR-100**

**Increasing \( L \) increases the performance of SEM.** We find that increasing \( L \), the number of simplex of SEM even beyond the over-complete regime increases the downstream accuracy. However, this increased performance is not observed when we abstain from using the softmax normalization of SEM. In Figure 5 using a ResNet-50 encoder, we compare BYOL + SEM, with an identical model without the Softmax normalization which we call BYOL + Embed. As, this is a control experiment, the extracted representation of BYOL + Embed is the embedder’s output \( z_{\theta} \). We fix \( V = 13 \) and scale \( L \in [10, 10000] \) to get a range of representation sizes. We observe that BYOL + Embed accuracy’s is decrease from probing \( z_{\theta} \) instead of \( c_{\theta} \) showing no gain from larger representations.

**Memory and computational efficiency of SEM.** We present the memory requirement and the computational efficiency of SEM in Table 5. The allocated memory represent the VRAM allocated by PyTorch during pre-training with a Batch Size of 256. As expected, SEM necessitates much more memory, which can become a practical issue. Fortunately, sparsiﬁying the matrix multiplication as discussed in Section 5.2 allows to considerably reduce the memory requirements while inducing a small reduction in performance. In term
of computational efficiency, we note that the cost of using SEM is small in comparison to the total cost of the pre-training and becomes marginal as we scale up the encoder.

![Graph](image)

**Table 5: Allocated memory, computation efficiency (calculated in FLOPs/sample) and accuracy of BYOL with SEM.**

<table>
<thead>
<tr>
<th>Resnet-18:</th>
<th>BYOL</th>
<th>BYOL+SEM</th>
<th>BYOL+SEM/4</th>
</tr>
</thead>
<tbody>
<tr>
<td>L=5000, V=13</td>
<td>50.7</td>
<td>70.5</td>
<td>70.5</td>
</tr>
<tr>
<td>L=1000</td>
<td>56.7</td>
<td>73.9</td>
<td>73.9</td>
</tr>
<tr>
<td>L=5000</td>
<td>56.7</td>
<td>73.9</td>
<td>73.9</td>
</tr>
</tbody>
</table>

Comparison of SEM with hard discretization approaches. Several other methods can be used to induce a sparse representation during pre-training and downstream classification. For example, we may sample L discrete one-hot codes each with V values using REINFORCE [Williams 1992] or Gumbel Straight-Through estimation [Jang et al., 2017] to back-propagate the gradient through the encoder. We could also use Vector Quantization (VQ) [Oord et al., 2018] as discussed in Liu et al. [2021] and consider L codebooks with V values each wherein the values are vectors in $\mathbb{R}^d$. However, when using these approaches instead of SEM, we see a considerable decrease in performance in comparison to the baseline as demonstrated in Table 6. In this table, we reproduce the same setup as SEM but we replace the Softmax with hard discretization baselines methods. For discretization with REINFORCE and 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 $\tau = 1$ for Gumbel Straight-Through. For VQ, we found that $L = 512$ and $V = 128$ led to the best performance. That is, we have 512 codebooks, each with 128 possible values that are represented by vectors in $\mathbb{R}^{32}$. We also present SEM with $\tau_D = 0$, which correspond to using the discretized representation for downstream classification, demonstrating that SEM with pre-training can be used to learn meaningful discrete codes for downstream task and yields better performance than the baselines, leading us to believe that SEM could be beneficial in setup where discretization is needed and pre-training is possible.

C.1 Analyzing the parameters of SEM

![Graph](image)

**Figure 4:** Effect of $\tau_p$ and $\tau_d$ on a RN-50. **Figure 5:** Comparing $f_{\text{SEM}}$ and $f_{\text{base}}$ on a RN-18.

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 $\tau_p$ and $\tau_d$ on the downstream accuracy. In Figure 5, we evaluate the effect of $L$ and $V$ on the downstream accuracy and also contrast $f_{\text{base}}$ and $f_{\text{SEM}}(\tau = 1)$, allowing us to confirm two predictions made in Theorem 1. The expected generalization improvement from SEM increases as we increase $L$ and as we increase $V$. We now discuss the effect of each of SEM’s parameter on the resulting downstream classification.

**Increasing V yields 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. In the regime that we can test it, larger $L$ seems to translate to better accuracy as observed in Figure 3 and Figure 5a.

The optimal $\tau_p$ 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. This is also observed in SEM. As presented in Figure 4a, the optimal $\tau_p$ for larger $V$ is higher.

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

D CIFAR100 superclass

The 100 classes of CIFAR-100 [Krizhevsky, 2009] are grouped into 20 superclasses. The list of superclass for each class in Table 7.

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

Table 7: Set of classes for each superclass on CIFAR-100.
E Additional CIFAR-100 coherence graphs

Figure 6: Comparison of the full semantic coherence graph $W_5$ between BYOL and BYOL + SEM.