# Localizing Memorization in SSL Vision Encoders

Wenhao Wang, Adam Dziedzic, Michael Backes, Franziska Boenisch

CISPA Helmholtz Center for Information Security

## TL;DR

We present the first work that allows to *localize* memorization in individual layers and units, i.e., neurons or convolutional filters, of self-supervised (SSL) vision encoders.

## Contributions

- A per-layer metric of memorization for SSL encoders (`LayerMem`), based on the `SSLMem` of our previous work.
- A per-unit metric of memorization for SSL encoders (`UnitMem`).
- Demonstration of the practical benefits of localization of memorization for encoder fine-tuning and pruning.
- Extensive empirical evaluation of `LayerMem` and `UnitMem` on various SSL frameworks and datasets.

## Summary of Findings

- Individual units in SSL encoders memorize individual training data points.
- SSL memorization increases with layer depth, highly memorizing units are distributed across the entire encoder.
- Units in SSL encoders experience significantly higher memorization of individual data points than units of models trained with supervised learning.
- In vision transformers, most memorization happens in the fully-connected layers.
- Atypical data points cause higher memorization in layers and units.

## Formalizing `LayerMem`

Recall our Definition of `SSLMem`:

$$\mathcal{H}_{\text{align}}(f,x,S) = \mathop{\mathbb{E}}_{f\sim\mathcal{A}(S)} \mathop{\mathbb{E}}_{x',x''\sim\text{Aug}(x)} [d\left(f(x'), f(x'')\right)]$$

$$\texttt{SSLMem}\ (g,f,x,S',S) = \text{H}_{\text{align}}(g,x,S') - \mathcal{H}_{\text{align}}(f,x,S)$$

The `LayerMem` for specific layer $l$ is defined as:

$$\texttt{LayerMem}(g,f,x,S',S,l) = \texttt{SSLMem}(g,f,x,S',S,l) - \texttt{SSLMem}(g,f,x,S',S,l-1)$$

## Formalizing `UnitMem`

We first define the mean activation $\mu$ of unit $u$ on a training point $x$ as:

$$\mu_u(x) = \mathop{\mathbb{E}}_{x'\sim\text{Aug}(x)} \text{activation}_u(x')$$

Further, for the unit $u$, we compute the maximum mean activation $\mu_{max,u}$ across all instances from $\mathcal{D}'$, where $N = |\mathcal{D}'|$, as

$$\mu_{max,u} = \max(\{\mu_u(x_i)\}_{i=1}^{N})$$

Let $k$ be the index of the maximum mean activation $\mu_u(x_k)$, *i.e.*, the *argmax*. Then, we calculate the corresponding mean activity $\mu_{-max}$ across all the remaining $N-1$ instances from $\mathcal{D}'$ as

$$\mu_{-max,u} = \text{mean}(\{\mu_u(x_i)\}_{i=1,i\neq k}^{N}).$$

Finally, we define the `UnitMem` of unit $u$ as

$$\texttt{UnitMem}_{\mathcal{D}'}(u) = \frac{\mu_{max,u} - \mu_{-max,u}}{\mu_{max,u} + \mu_{-max,u}}.$$

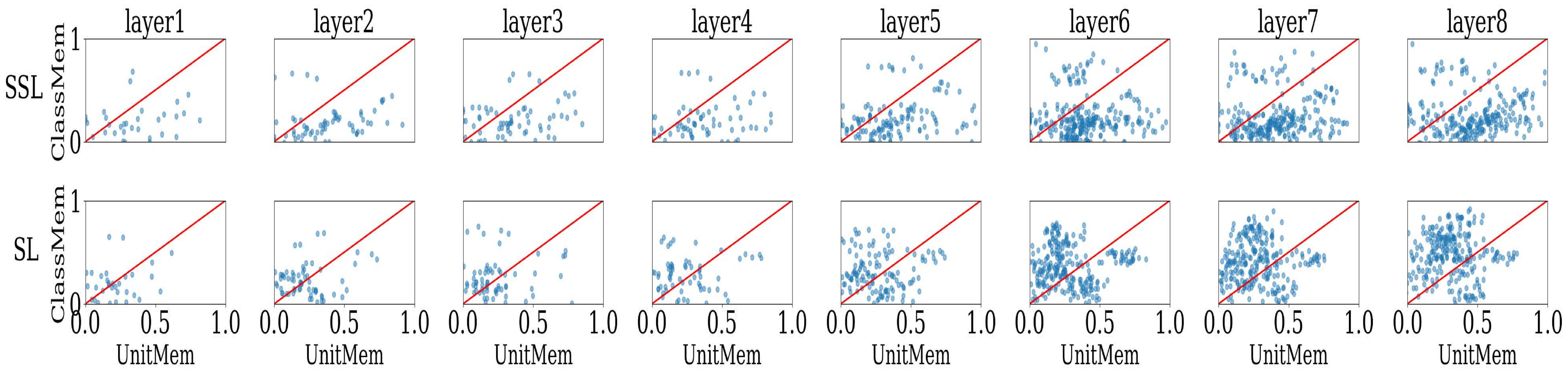
## Insights into `LayerMem` & `UnitMem`

| ViT Block Number | *Attention Layer* `LayerMem` | `ΔLayerMem` | `Res Block` | *Fully-Connected Layer* `LayerMem` | `ΔLayerMem` | `Res Block` |
|---|---|---|---|---|---|---|
| 1 | 0.006 | - | 0.007 | 0.020 | - | 0.022 |
| 2 | 0.028 | 0.006 | 0.028 | 0.039 | 0.011 | 0.040 |
| 3 | 0.046 | 0.006 | 0.047 | 0.060 | 0.013 | 0.061 |
| 4 | 0.067 | 0.006 | 0.067 | 0.083 | 0.017 | 0.085 |
| 5 | 0.092 | 0.007 | 0.091 | 0.105 | 0.014 | 0.106 |
| 6 | 0.114 | 0.008 | 0.114 | 0.129 | 0.015 | 0.131 |
| 7 | 0.140 | 0.009 | 0.139 | 0.155 | 0.016 | 0.156 |
| 8 | 0.164 | 0.008 | 0.164 | 0.182 | 0.018 | 0.182 |
| 9 | 0.191 | 0.009 | 0.190 | 0.210 | 0.020 | 0.211 |
| 10 | 0.220 | 0.009 | 0.220 | 0.240 | 0.020 | 0.241 |
| 11 | 0.249 | 0.008 | 0.249 | 0.271 | 0.022 | 0.271 |
| 12 | 0.280 | 0.009 | 0.280 | 0.303 | 0.023 | 0.304 |

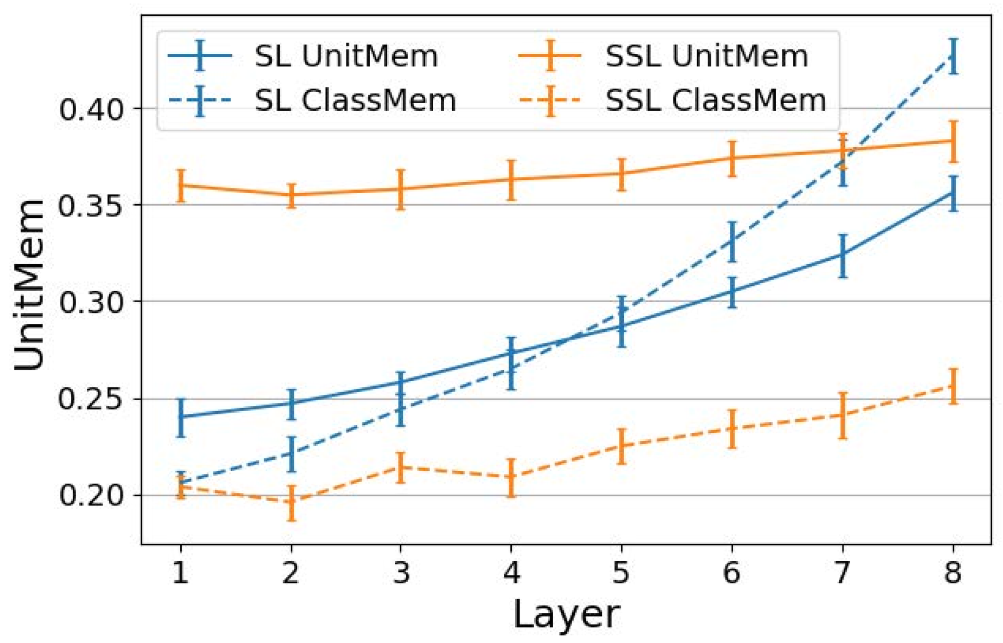
**The memorization in ViT occurs mainly in the deeper blocks and more in the fully-connected than attention layers.**

| Layer | `LayerMem` | ΔLM | `LayerMem` Top50 | ΔLM Top50 | `LayerMem` Least50 |
|---|---|---|---|---|---|
| 1 | 0.091 | - | 0.144 | - | 0.003 |
| 2 | 0.123 | 0.032 | 0.225 | 0.081 | 0.012 |
| 3 | 0.154 | 0.031 | 0.308 | 0.083 | 0.022 |
| 4 | 0.183 | 0.029 | 0.402 | 0.094 | 0.031 |
| Res2 | 0.185 | 0.002 | 0.403 | 0.001 | 0.041 |
| 5 | 0.212 | 0.027 | 0.479 | 0.076 | 0.051 |
| 6 | 0.246 | 0.034 | 0.599 | 0.120 | 0.061 |
| 7 | 0.276 | 0.030 | 0.697 | 0.098 | 0.071 |
| 8 | 0.308 | 0.032 | 0.817 | 0.120 | 0.073 |
| Res6 | 0.311 | 0.003 | 0.817 | 0 | 0.086 |

**Memorization Increases with layer depth but not Mono-tonically.**




**Highly memorized units occur over whole encoder.**




**SL have higher `ClassMem` while SSL have higher `LayerMem`.**

Github Repo

ArXiv Paper

## Application of `LayerMem` & `UnitMem`

| Fine-tuned Layers | Accuracy (%) ↑ |
|---|---|
| None (HEAD) | 48.6% ± 1.12% |
| 6 (highest ) + HEAD | **53.0% ± 0.86%** |
| 8 (last layer, highest ) + HEAD | 52.7% ± 0.97% |
| 6,8 + HEAD | **56.7% ± 0.84%** |
| 7,8 + HEAD | 55.3% ± 0.77% |
| 4,6,8 (highest ) + HEAD | **57.9% ± 0.79%** |
| 6,7,8 + HEAD | 56.5% ± 0.95% |

**Fine-tuning most memorizing layers according to `LayerMem`.** ResNet9 encoder trained with SimCLR on CIFAR10 and fine-tune different (combinations of) layers on the STL10 dataset

| Pruning Strategy | % of Selected Units | *Downstream Accuracy (%)* CIFAR10 | SVHN | STL10 |
|---|---|---|---|---|
| No Pruning | - | 70.44 | 78.22 | 69.12 |
| Top per layer | 10 | 53.04 | 63.84 | 50.94 |
| Random per layer | 10 | 58.09 ± 1.76 | 67.04 ± 2.44 | 55.71 ± 2.18 |
| Low per layer | 10 | 62.58 | 72.26 | 59.26 |
| Top per layer | 20 | 48.30 | 55.88 | 43.18 |
| Random per layer | 20 | 51.34 ± 1.21 | 58.01± 1.34 | 46.74 ± 0.97 |
| Low per layer | 20 | 54.84 | 62.60 | 50.02 |
| Top total | 10 | 49.16 | 61.28 | 47.30 |
| Random total | 10 | 56.77 ± 2.09 | 67.09 ± 1.56 | 53.89 ± 2.33 |
| Low total | 10 | 62.62 | 72.28 | 59.30 |

**Removing the least/most memorized units according to `UnitMem` preserves most/least linear probing performance.**

## Conclusions

- We propose the first practical metrics for localizing memorization within SSL encoders on a per-layer (`LayerMem`) and per-unit (`UnitMem`) level.
- While memorization in SSL increases in deeper layers, a significant fraction of highly memorizing units can be encountered over the entire encoder.
- SSL encoders significantly differ from models trained with supervised learning in their memorization patterns, with the former constantly memorizing data points and the latter increasingly memorizing classes.