Learning from Memory: Non-Parametric Memory Augmented Self-Supervised Learning of Visual Features

Thalles Silva [1] Helio Pedrini [1] Adín Ramírez Rivera [2]

Abstract

This paper introduces a novel approach to improving the training stability of self-supervised learning (SSL) methods by leveraging a non-parametric memory of seen concepts. The proposed method involves augmenting a neural network with a memory component to stochastically compare current image views with previously encountered concepts. Additionally, we introduce stochastic memory blocks to regularize training and enforce consistency between image views. We extensively benchmark our method on many vision tasks, such as linear probing, transfer learning, low-shot classification, and image retrieval on many datasets. The experimental results consolidate the effectiveness of the proposed approach in achieving stable SSL training without additional regularizers while learning highly transferable representations and requiring less computing time and resources. Code at https://github.com/sthalles/MaSSL.

1. Introduction

Self-supervised learning (SSL) based on join-embedding architectures currently holds state-of-the-art performance on many representation learning benchmarks. Among different methods, clustering-based approaches (Caron et al., 2021; 2018; 2019; Asano et al., 2019; Silva & Ramírez Rivera, 2023) appear to be the most successful recipe for learning self-supervised features in the visual domain. SSL clustering methods learn image representations by discretizing the embedding space. They set up optimization tasks that involve learning a finite set of prototypes or centroids based on the self-supervised signal coming from views of an image. Despite their high ability to learn representations, clustering methods are notoriously difficult to train due to their susceptibility to training collapse.

We argue that learning the prototypes via gradient descent is the primary source of poor training instability in SSL clustering methods. Due to the lack of human labels and excessive noise from the self-supervised signals, the network attempts to cluster all the embeddings into a single prototype as the most efficient way to optimize the loss function. Current methods avoid collapse by employing additional regularizers that force representations to spread evenly in the space over the prototypes. Examples include: (1) the combination of centering and target sharpening (Caron et al., 2021; Zhou et al., 2022), (2) the mean entropy maximization (ME-MAX) (Assran et al., 2021; Silva & Ramírez Rivera, 2023), and (3) the Sinkhorn-Knopp (Asano et al., 2019; Caron et al., 2020). In addition, state-of-the-art SSL methods based on Vision Transformers (ViTs) use the full output of the Transformer (`[CLS]` + patch token) and optimize the MIM (Masked Image Modeling) pretext task on the patch embeddings (Zhou et al., 2022; Oquab et al., 2023). Despite performance gains, this architectural choice drastically increases computational costs and training time.

Motivated by the current landscape of SSL methods, we propose a new stable method that exceeds current approaches on many retrieval and transfer tasks while reducing computing resources and training time. Based on the intuition that learning relies on memory, we present a non-parametric approach that poses the SSL problem in terms of learning from past experiences. We augment a neural network with a memory component that holds a snapshot of the most recent image representations seen by the model. Unlike previous approaches that use memory/queues to mine negatives (He et al., 2020) or positives (Dwibedi et al., 2021) in a contrastive learning setup, our proposed memory allows the network to learn visual representations by comparing current events (views of an image) with previously experienced concepts (image representations from previous iterations) in memory. We named this method **M**emory **A**ugmented **S**elf-**S**upervised **L**earning (MaSSL).

In addition to the working memory, we introduce the concept of stochastic memory blocks. Stochastic blocks allow

[1] Institute of Computing, University of Campinas, Campinas-SP, Brazil [2] Department of Informatics, University of Oslo, Oslo, Norway. Correspondence to: Thalles Silva <thalles.silva@students.ic.unicamp.br>, Helio Pedrini <helio@ic.unicamp.br>, Adín Ramírez Rivera <adinr@uio.no>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

the network to retrieve a random subset of representations from previous iterations. These representations symbolize concepts previously seen by the model and are used as anchors to enforce consistency between the current image views. We show that stochastic memory blocks regularize the learning problem, making our method stable even without additional regularizers to prevent mode collapse. Finally, our loss optimizes for consistency between views of an image by matching their view-memory similarity distributions, which means that views of an image must activate similar memory representations with similar scores. In other words, views should output consistent similarity patterns when compared to representations of other images in the memory. Figure 1 presents a pictorial overview of our learning architecture.

Our contributions are threefold:

- A novel SSL pretext task that learns visual representations by formulating multiple discriminative tasks based on comparing the current perceived signal to previously experienced concepts stored in memory.
- A stochastic memory, implemented through a non-parametric distribution of the past image representations and a memory block mechanism that allows representation learning in a prototype-free manner.
- A simple SSL learning framework that does not require additional regularizers to avoid training collapse and operates on the [CLS] token of the ViT, reducing the pretraining time and memory requirements while learning highly transferable representations.

2. Related Work

Self-supervised learning has evolved from more specialized pretext tasks such as solving rotations (Gidaris et al., 2018), jigsaw puzzles (Noroozi & Favaro, 2016), and relative positions (Doersch et al., 2015), to a predominant set of tasks based on instance discrimination (He et al., 2020; Chen et al., 2020; Chen & He, 2021; Silva et al., 2023). Current methods mainly differ from one another on (1) how they avoid mode collapse and (2) how they pose the view-invariance task, which may be embedding- (Grill et al., 2020) or clustering-based (Caron et al., 2020). Current state-of-the-art SSL is based on the Transformer architecture (Dosovitskiy et al., 2020). Some approaches formulate their loss function over the [CLS] token only (Caron et al., 2021), while the most recent and powerful methods use the full output of the Transformer, i.e., [CLS] + patch tokens (Zhou et al., 2022; Oquab et al., 2023).

Memory banks or queues in SSL are not new concepts. Many proposed techniques (Misra & Maaten, 2020; Chen et al., 2021) rely on storing representations in a container, often called support set or queues. In MoCo (Chen et al., 2021) and PIRL (Misra & Maaten, 2020), the memory is used as a source of negative representations, i.e., the currently processed image is pushed away from distinct image representations in a queue in a contrastive task by minimizing the InfoNCE (Oord et al., 2018) loss. Alternatively, Dwibedi et al. (2021) uses an extra queue as a source of positives. Specifically, the support set is used to bootstrap nearest neighbors for the current views, framing a contrastive learning task that approximates views of an image to their neighbors' representations.

MaSSL uses the memory container differently. To begin, MaSSL is a negative-free contrastive[1] method. Hence, it does not need to bootstrap negatives for its learning objective, nor does it have an explicit term in the loss function to push representations apart and avoid collapse. Most importantly, MaSSL uses the memory to formulate discriminative tasks. Intuitively, if the currently perceived image is similar to one or more images the model has seen (in memory), they should relate with a strong similarity score. Conversely, if the current image is not semantically similar to one or more images in the memory, they should relate with a weak similarity score. On top of that, MaSSL's learning objective forces multiple views of the same image to agree on how they relate to previously perceived representations.

Similar to MaSSL, Assran et al. (2021) proposed a semi-supervised method, termed PAWS, that employs a support set composed of uniformly distributed labeled examples as anchors to optimize for views' consistency. While MaSSL may be regarded as an SSL version of PAWS, translating the learning problem in the former to an SSL setup is not trivial. PAWS takes advantage of the additional, *free-of-noise* signal to incorporate biases into the learning problem, stabilizing the training process. In addition to human labels, PAWS uses a regularizer to spread views' assignments over the examples in the support set. In contrast, MaSSL does not require human-labeled examples to learn visual representations, nor does it employ regularizers to prevent collapse.

3. Methodology

Inspired by how humans recall and generalize observations based on memory comparisons, we introduce a memory (based on a non-parametric distribution) to our methods, allowing the network to contrast representations from current events to previous iterations. In addition, we regularize the learning process using a stochastic memory partition strategy, forcing the representations to be general and not susceptible to particular shortcuts. Figure 1 depicts an overview of our proposed method.

[1] The literature uses the misnomer "non-contrastive" to refer to methods that do not explicitly use negative examples while learning the representations. We, however, argue that there is a better way of naming these methods.

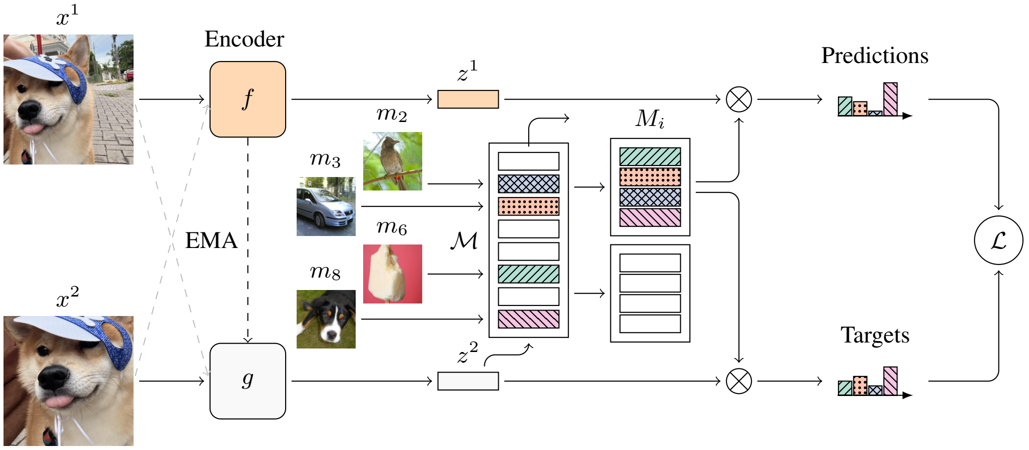

Figure 1. Learning from memory. Given two or more views of an image, each view is encoded by the student and teacher encoders, resulting in respective vector representations z^1 and z^2. Each view's representation is compared against representations of previously seen images in memory, resulting in respective similarity distributions. Note that the working memory $\mathcal{M}$ is split into blocks, M_i, of randomly chosen representations. The learning objective, $\mathcal{L}$, forces the similarity distributions of views w.r.t. the memory representations to be consistent. In a case where the model perceives an image of a dog, the interaction between what it currently sees and what it remembers should produce (1) strong similarity scores for previously seen dogs, (2) weak scores for non-related images in the memory, and (3) interactions should be consistent among views.

Notation. Let x be an image from a large unlabeled dataset $\mathcal{X}$ and x^v a view of x, for $v = 1, 2, \dots, N_v$. We define f and g as a pair of student and teacher vision encoders where the student is trained with backpropagation, and the teacher is distilled through exponential moving average (EMA) from the student. Each encoder receives a view, x^v, and produces a low-dimensional representation z^v, such that $z_s^v = f(x^v)$ and $z_t^v = g(x^v)$, where the subscripts s and t represent student and teacher branches.

In addition, we denote by $\mathcal{M} = \{m_k\}_{k=0}^K$ a vector container designed to simulate a working memory that temporarily stores vector representations from images previously seen during training by the model.

3.1. Memory

When humans experience something for the first time, there is likely an additional excitement or surprise due to encountering novelty. When a similar experience happens again, however, the surprise will probably not be the same. This occurs because of memory and its essential role in learning. Indeed, the fact that humans can recognize the first time hearing or seeing something is a testament to the fact that we are constantly comparing what we perceive with previous experiences to make sense of the world around us.

At a high level, memory allows for three crucial processes: (1) acquisition of new information (encoding), (2) information retention over time (storage), and (3) retrieval. Through these processes, we can make sense of our present and take informed actions based on past observations.

Our learning framework explores such characteristics of memory. Given a pair of views x^1 and x^2, while most SSL methods compare views directly or against learnable prototypes, we seek to design a task that forces the neural network to utilize previously experienced concepts as discriminative cues to learn representations that are invariant to view changes. This task (3) must produce consistent predictions for different views of an image x based on the similarity perspectives of representation vectors stored in memory. In other words, a pair of views x^1 and x^2 must have consistent similarity relationships to previous concepts experienced by the model.

In practice, the memory is a non-parametric distribution that stores encoded representations $z^{(\cdot)}$ from the current batch of image views. To update the memory, we implement a FIFO protocol (First-In, First-Out), i.e., representations enter from one end of the memory and are discarded from the other. This storage pattern preserves an *ordering* bias in which one end of the memory holds recently updated representations while the other holds older ones. Since representations constantly evolve during training, this ordering bias could drive the learning algorithm to give more weight to the recently remembered representations stored in one end of the memory. We break the ordering dependency by introducing a stochastic component when retrieving representations from memory. We show that such a strategy regularizes the model and improves representations, cf. Section 5.3.

3.2. Optimizing over Random Memory Blocks

Inspired by Silva & Ramírez Rivera's (2023) work on the random partition pretext task, we empirically found that applying a similar principle to our proposed memory component to break it into multiple disjoint subsets further improves performance and training stability, cf. Section 5.3. Randomizing the memory representations into independent smaller blocks effectively mitigates the ordering bias that arises from inserting recent experiences into one end of the FIFO memory. This approach not only improves the overall performance but also enhances the training stability of the system. Consequently, we transition from a single task over all representations in memory (ordered by insertion time) to a series of smaller tasks, each operating on a small subset of independent memory representations.

Let $\mathbb{M} = \{M_1, M_2, \dots, M_B\}$ form a partition of the set $\mathcal{M}$, where M_b, for $b = 1, 2, \dots, B$ is a non-empty subset, a memory block containing randomly chosen representations sampled from $\mathcal{M}$.

The framework starts by computing the representation vectors $z^1 = f\left(x^1\right)$ and $z^2 = g\left(x^2\right)$, independently, for each view x^v. Then, we retrieve a memory block M_b (a random subset from $\mathcal{M}$) and compute the similarity scores between the views and the memory block as

$$p_s^1 = \operatorname{softmax}\left(\cos\left(z^1, M_b\right)/\tau_s\right), \tag{1}$$

$$p_t^2 = \operatorname{softmax}\left(\cos\left(z^2, M_b\right)/\tau_t\right), \tag{2}$$

where $\cos\left(\cdot,\cdot\right)$ is the cosine similarity function, τ_s and τ_t are student and teacher temperature hyper-parameters, and p_s^1 and p_t^2 are the *view-memory similarity relationship* obtained from comparing the views' representations z^1 and z^2 and the representations m_k within a memory block M_b.

Intuitively, the term $\cos\left(z^{(\cdot)}, M_b\right)$ compares what is being perceived at the moment, i.e., the current image views $x^{(\cdot)}$, with what has been experienced in the past, M_b, creating a view-memory similarity relationship $p^{(\cdot)}$. Once we contrast the current views' and memory blocks' representations, we force the view-memory similarity relationship to be consistent using a regular cross-entropy loss, such as

$$\mathcal{L}_b\left(p_s^1, p_t^2\right) = -p_t^{2,b}\log\left(p_s^{1,b}\right), \tag{3}$$

where the subscript b indexes the memory blocks M_b. The overall loss is the aggregation over the memory block losses, i.e., $\mathcal{L} = \sum_b \mathcal{L}_b$.

Optimizing the loss function (3) forces a consistent assignment of views from the perspective of the representations currently remembered by the model. Given a memory block M_b of size N_b, the problem can be seen as a N_b-way classification task where each representation in a memory block represents a different semantic perspective. This way, to achieve consistency between the pair of views, the similarity relationship between what the network remembers and the different views of an image must be consistent. Intuitively, if we consider C as the number of hidden classes in $\mathcal{X}$ and that the memory $\mathcal{M}$ is large enough $K \gg C$ such that we can assume the memory holds a fair number of representations from each hidden category, we would strive for two main properties when optimizing the loss function (3): (1) the view-memory similarity relationship of each view should be consistent and (2) the view-memory similarities should be higher for cases where the current views and the recalled representations share semantic structure, i.e. remembering from a previously experienced concept.

4. Main Results

We assess MaSSL's representations on a broader set of computer vision benchmarks, focusing on the challenging scenario of transfer learning with frozen features.

4.1. Transfer Learning

We follow the transfer learning evaluation protocol proposed by Silva & Ramírez Rivera (2023) based on k-NN. We validate MaSSL's representations on *eight* datasets across four different values of k and report results on Table 1. For a fixed value of $k = 20$, MaSSL's **achieve better transfer scores on five out of the eight (5/8) datasets, with k-NN performance gains of +2.6 and +4.6 on AirCraft and GTSRB datasets respectively**. On average, over all datasets, MaSSL outperforms competitors with **performance gains of nearly +2.5 for all values of k**. We report additional experiments on Table A.1 in Appendix A.1.

In addition to the non-parametric k-NN benchmark, we train logistic regression classifiers on top of the frozen features of the pre-trained ViT-B encoder. In Table 2, we compare the performance of SSL methods on *six* datasets. MaSSL's representations **achieve higher transferable scores in four of the six datasets (4/6)**, highlighting the high transfer-learning power of MaSSL's pre-trained representations.

4.2. Linear Evaluation

In Table 3, we report in-domain linear evaluations for ViT-S/B backbones by training linear models with SGD, cf. Appendix A.2 for details on the protocol. Additionally, we report k-NN performance on the full ImageNet-1M. MaSSL performs on par with iBOT on both metrics, with a slight performance gain on ViT-B. We report the supervised baseline from Touvron et al.'s (2021) work for reference.

4.3. Image Retrieval Benchmark

Following previous work (Caron et al., 2021; Zhou et al., 2022), we evaluate MaSSL's pre-trained representations on retrieval tasks based on *landmark* and *copy detection*.

Image Retrieval. We consider the widely used revisited Oxford-5k and Paris-6k image retrieval datasets (Raden-

Table 1. **Transfer learning k-NN evaluation.** We report top-1 accuracy ($k = 20$) for individual datasets and averages over all datasets for $k \in \{10, 20, 100, 200\}$. Results for ViT-B/16.

		Pets	Flowers	Aircraft	Cars	Country	Food	STL	GTSRB	Avg @k			
Methods	Epo.	results for $k = 20$								10	20	100	200
MAE	800	19.4	16.9	9.7	6.0	5.0	11.9	64.6	27.6	20.9	20.1	16.9	15.2
MoCo-v3	300	83.8	70.2	27.4	22.4	14.3	64.5	97.5	56.1	55.3	54.5	52.2	51.3
DINO	800	90.1	84.6	38.5	32.7	**15.9**	70.7	98.9	64.7	62.0	62.0	60.8	60.2
iBOT	800	89.2	83.4	33.7	28.8	15.7	**72.6**	**99.0**	63.0	60.8	60.7	59.5	58.8
Ours	800	**91.6**	**84.6**	**41.1**	**33.3**	15.7	72.5	98.8	**69.3**	**63.3**	**63.4**	**62.4**	**61.8**

Table 2. **Transfer learning with logistic regression.** We report top-1 accuracy for logistic regression models trained on top of the frozen features of SSL ViTs pre-trained on the ImageNet-1M.

Method	DTD	C100	GTSRB	Cars	Air	Pets
DINO	**73.3**	82.6	86.8	70.3	66.1	93.5
iBOT	71.8	**84.0**	85.8	70.5	64.5	93.8
Ours	71.7	83.1	**89.1**	**73.7**	**67.2**	**94.2**

Table 3. **Linear probing and k-NN evaluation on ImageNet-1M.**

Method	Arch	Epo.	Linear	k-NN
MoCo-v3	ViT-S/16	300	73.4	–
DINO	ViT-S/16	300	76.2	72.8
iBOT	ViT-S/16	300	77.4	74.6
Ours	ViT-S/16	300	**77.5**	**74.7**
DeiT (Sup)	ViT-S/16	800	79.8	–
DINO	ViT-S/16	800	77.0	74.5
iBOT	ViT-S/16	800	**77.9**	**75.2**
Ours	ViT-S/16	800	77.8	75.1
DeiT (Sup)	ViT-B/16	400	81.8	–
MoCo-v3	ViT-B/16	400	76.7	–
NNCLR	ViT-B/16	1000	76.5	–
DINO	ViT-B/16	400	78.2	76.1
iBOT	ViT-B/16	400	79.5	77.1
Ours	ViT-B/16	400	**79.6**	**77.2**

Table 4. **Image retrieval.** We report mAP on the revisited Oxford-5k and Paris-6k retrieval datasets for different SSL methods using pre-trained frozen features from different ViT backbones.

			$\mathcal{RO}$x		$\mathcal{R}$Par	
Method	Arch	Epo.	M	H	M	H
Sup	RN101+R-MAC	100	49.8	18.5	74.0	52.1
MoCo-v3	ViT-S/16	300	21.7	5.1	38.9	13.1
DINO	ViT-S/16	800	37.2	13.9	63.1	34.4
iBOT	ViT-S/16	800	36.6	13.0	61.5	34.1
Ours	ViT-S/16	800	**38.5**	**15.9**	**63.4**	**34.8**
MoCo-v3	ViT-B/16	300	30.5	8.6	54.3	23.5
DINO	ViT-B/16	400	37.4	13.7	63.5	35.6
iBOT	ViT-B/16	400	36.8	**14.3**	64.1	36.6
Ours	ViT-B/16	400	**39.3**	14.1	**65.8**	**38.1**

Table 5. **Copy detection.** We report mAP on the "strong" subset of the INRIA Copydays using frozen features from pre-trained ViTs.

Method	Arch	Epo.	mAP
DINO	VIT-S/16	800	**85.7**
iBOT	VIT-S/16	800	83.7
Ours	VIT-S/16	800	85.5
DINO	VIT-B/16	400	86.8
iBOT	VIT-B/16	400	84.2
Ours	VIT-B/16	400	**87.6**

ović et al., 2018), containing 3 distinct sets of increasing difficulty, each with query/database pairs. In Table 4, we report mAP (mean average precision) on the Medium (M) and Hard (H) sets, ensuring fair comparisons with previous work. For reference, we report the performance of a supervised method (Revaud et al., 2019) tailored for image retrieval tasks.

Among SSL methods, DINO is a strong baseline and beats iBOT in most instances. MaSSL's ViT-S surpasses DINO in all scenarios while our ViT-B pre-trained encoder outperforms DINO in **three out of the four (3/4) scenarios**, only losing in the Hard set of the Oxford-5k dataset by -0.2.

Copy Detection. In addition, we consider the INRIA Copydays dataset (Douze et al., 2009) for evaluation on the copy detection task. Following Zhou et al.'s (2022) protocol, we report mAP on the "strong" subset without additional distractors in Table 5. For ViT-B, MaSSL increases upon the baseline performance from DINO and iBOT by **+0.8%** and **3.4%**, respectively.

4.4. Low-Shot and Long-Tailed Evaluation

In Table 6, we assess pre-trained representations on learning from a few labeled examples, where we consider 1% and 10% of the ImageNet labels. Moreover, we measure the impact of using different evaluation protocols on low-shot classification by employing a non-parametric k-NN, a logistic regression estimator, and a linear model (single layer MLP) trained with SGD. **MaSSL outperforms DINO and iBOT in most setups**. Interestingly, for ViT-S, training a linear model with SGD tends to underperform compared to logistic regression or even k-NN. However, when more data or a more complex encoder is used, k-NN acts as a lower bound, while logistic regression and MLP alternate as the most effective evaluators.

Table 6. **Low-shot classification on ImageNet-1M.** Evaluations on three protocols (k-NN, 1-layer MLP, and logistic regression) and two data regimes (1% and 10% of ImageNet-1M labels).

Method	Arch	Protocol	1%	10%
DINO	ViT-S/16	k-NN	61.3	69.1
iBOT	ViT-S/16	k-NN	62.5	70.1
Ours	ViT-S/16	k-NN	**62.6**	**70.4**
DINO	ViT-S/16	Linear	60.5	71.0
iBOT	ViT-S/16	Linear	**61.5**	72.6
Ours	ViT-S/16	Linear	61.4	**72.6**
DINO	ViT-S/16	LogReg	64.5	72.2
iBOT	ViT-S/16	LogReg	65.9	**73.4**
Ours	ViT-S/16	LogReg	**65.9**	73.2
DINO	ViT-B/16	k-NN	62.5	70.1
iBOT	ViT-B/16	k-NN	66.3	72.9
Ours	ViT-B/16	k-NN	**68.8**	**74.1**
DINO	ViT-B/16	Linear	66.2	74.2
iBOT	ViT-B/16	Linear	68.2	75.7
Ours	ViT-B/16	Linear	**70.4**	**76.4**
DINO	ViT-B/16	LogReg	67.1	74.2
iBOT	ViT-B/16	LogReg	69.6	75.9
Ours	ViT-B/16	LogReg	**71.3**	**76.3**

Table 7. **Low-shot and long-tailed evaluations.** We report Top-1 accuracy for ViT-B/16 on low-shot and long-tailed ImageNet.

	# images per class			ImNet-LT
	1	2	4	top-1
MoCo-v3	37.7± 0.3	47.8± 0.6	54.8± 0.2	56.7
DINO	39.2± 0.4	49.3± 0.8	57.6± 0.4	63.7
iBOT	42.2± 0.7	52.8± 0.3	60.6± 0.3	66.2
Ours	**44.8± 0.4**	**56.3± 0.3**	**63.8± 0.2**	**67.9**

In Table 7, we consider long-tailed learning and challenging low-shot scenarios. We train linear models using frozen features on the ImageNet-LT dataset (Liu et al., 2019), which is a highly unbalanced version of the ImageNet-1M. We report top-1 accuracy on the ImageNet-LT *balanced* test set. In addition, we report top-1 accuracy on balanced subsets of ImageNet-1M containing *one*, *two*, and *four* randomly chosen examples per class. MaSSL shows **significant learning efficiency on extreme low-shot scenarios and robustness to highly unbalanced data**. Cf. Appendix A.3 for more details.

4.5. Robustness Evaluation

Vision models rely on foreground and background information when classifying objects in images. Even when the correct object is present in an image, changes in the background may cause the network to classify the object incorrectly. To understand how background-robustness in SSL methods, we follow the protocol from Zhou et al. (2022) and assess the robustness of pre-trained SSL representations against background changes using the ImageNet-9 (IN-9) dataset (Xiao et al., 2020). The IN-9 evaluation protocol masks/superimposes foregrounds on adversarially chosen backgrounds to define the following seven variants: Only-FG (OF), Mixed-Same (MS), Mixed-Rand (MR), Mixed-Next (MN), No-FG (NF), Only-BG-B (OBB), and Only-BG-T (OBT). In the first four variants, the original foreground is kept while the background is modified. In the last three variants, the original foreground is masked.

Table 8. **Robustness evaluation against background changes.** ViT-B results on the IM-9 dataset over 7 variants of foreground/background mixing and masking.

	Background changes							Clean
	OF	MS	MR	MN	NF	OBB	OBT	IN-9
iBOT	91.9	89.7	81.9	79.7	54.7	17.6	20.4	96.8
Ours	91.0	**90.2**	**83.0**	**80.4**	53.4	15.8	**23.7**	**97.6**

In Table 8, we report results for ViT-B backbones trained for 400 epochs and then evaluated using a linear head for 100 epochs. Even though MaSSL only trains on the `[CLS]` token of the ViT, it still surpasses iBOT, which performs MIM (Masked Image Modeling) on the patch tokens, **on four out of the seven (4/7) variants**.

Table 9. **Clustering evaluation**. We report (NMI) normalized mutual information, (AMI) adjusted mutual information, and (ARI) adjusted rand index.

		ImageNet-1%			CIFAR-10		
Method	Arch	NMI	AMI	ARI	NMI	AMI	ARI
DINO	RN-50	69.2	46.2	21.7	39.6	39.5	28.0
CARP	RN-50	70.3	48.0	23.9	49.0	48.9	38.7
DINO	ViT-B/16	79.1	64.3	38.1	58.7	**58.5**	**27.4**
iBOT	ViT-B/16	81.3	68.1	42.0	57.7	57.5	26.8
Ours	ViT-B/16	**81.7**	**68.7**	**44.1**	**58.7**	58.4	27.2

4.6. Clustering Evaluation

In addition to the supervised evaluations in Section 4.2, we assess pre-trained representations using unsupervised metrics on the ImageNet-1% and CIFAR-10 datasets in Table 9. MaSSL **outperforms iBOT in all cases and performs comparably to DINO**.

4.7. Training Time and GPU Memory

One important advantage of MaSSL over other SSL methods based on ViTs is the trade-off between training resources (plus time) and performance. DINO and iBOT learn prototypes using gradient descent. DINO trains 65 536 prototypes, which translates into 16 777 216 extra trainable parameters, given the standard representation vector dimension of 256. On the other hand, MaSSL avoids learning prototypes from scratch and implements a stochastic non-parametric memory component using representations from previous iterations, which require negligible extra comput-

Table 10. **Training time and memory.** We compare performance (k-NN on ImageNet-1M), training time (hours), and memory requirements (Gigabytes) for SSL methods based on ViT-S/16 backbones pre-trained with a global batch size of 1024 images.

	100 epochs		300 epochs		800 epochs		
	k-NN	Time	k-NN	Time	k-NN	Time	Mem
DINO	69.7	24.2h	72.8	72.6h	74.5	180.0h	15.4G
iBOT	71.5	24.3h	74.6	73.3h	75.2	193.4h	19.5G
Ours	72.7	24.2h	74.7	72.4h	75.1	177.3h	15.1G

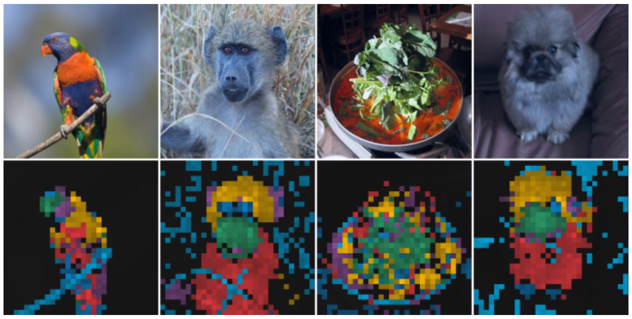

Figure 2. Visualization of MaSSL's self-attention maps. Multiple heads are displayed in different colors.

ing memory since it does not receive gradient updates. As shown in Table 10, training MaSSL for 800 epochs using a ViT-S backbone is nearly 9% faster, requires 25.6% less memory, achieves comparable linear probing on ImageNet-1M and better transfer performance on many datasets.

iBOT trains two sets of prototypes, each containing 8192 output neurons, and requires 4 194 304 trainable parameters to learn the prototype layers. iBOT also uses the full output of the transformer, i.e., the [CLS] plus *patch tokens*, which drastically increases its memory footprint and training time. Differently, MaSSL only trains on the [CLS] token of the ViT, still delivering good transferable performance in less time and with less memory. Overall, MaSSL **achieves the best trade-off between performance and training resources**. All methods were trained on two 8-GPU V100 machines with a batch size of 1024.

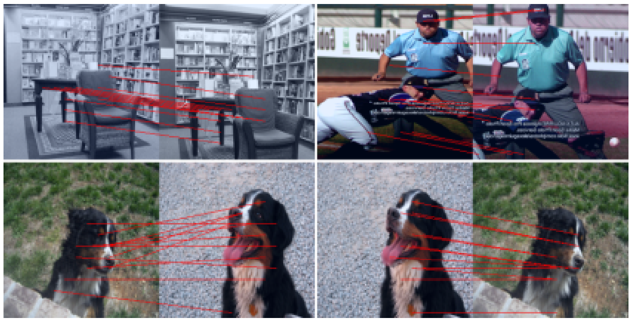

Figure 3. Sparse correspondence results for MaSSL.

4.8. Visualizing Self-Attention Maps

To analyze the internal representations of MaSSL and to understand its powerful retrieval properties, we follow the protocols of Caron et al. (2021); Zhou et al. (2022) and visualize the attention maps of MaSSL's pre-trained ViT-S encoder. The [CLS] token is used as the query vector, and the visualizations are from different heads of the last layer displayed in various colors.

In Figure 2, we see the high attentive capabilities of MaSSL. For instance, in the first column, we see individual heads paying attention to different portions of the image, such as the bird's beak, different parts of its body, and the wood. In the second column, we can visually isolate the monkey's top head, body, and face. In the third column, multiple heads attend to different parts of the food. We present a detailed visual overview of MaSSL's self-attention layers in Appendix C.1.

4.9. Sparse Correspondence

In Figure 3, we evaluate MaSSL's performance on the sparse correspondence task proposed by Zhou et al. (2022). The task is to match patch representations from two images. In the first row of Figure 3, we match patches from two views of the same image, while in the second row, we match patches from two distinct images of the same class. Interestingly, even though MaSSL does not train at patch-level representations, as iBOT, it still performs surprisingly well on dense prediction tasks such as sparse correspondence. We provide additional visualizations in Appendix C.2.

5. Ablations

In this section, we investigate the main components of MaSSL. Unless otherwise specified, ablations experiments are pre-trained for 100 epochs using the ViT-S architecture varying hyperparameters according to the experiments.

5.1. The Effect of the Memory Size

One natural aspect of the proposed memory component in Section 3.1 is how its size affects the learned representation. Intuitively, a large memory retains information for longer, allowing the model to compare the current image views to a broader distribution of remembered concepts. Also, for fixed block sizes N_b, a large memory M allows for more blocks, increasing signal processing during training. On the other hand, a smaller memory reduces the span and distribution of stored concepts.

In Table 11, we investigate the effect of the memory size on MaSSL' performance. We fix the memory block size as $N_b = 4096$ and vary the memory size K. We report top-1 accuracy using k-NN. Experiments suggest that a

Table 11. A larger memory benefits the learned representations.

K	8192	16384	32768	65536	131072
k-NN	67.8	69.9	70.5	71.9	71.9

Table 12. Larger block sizes N_b benefit the learned representations.

N_b	512	1024	2048	4096	8192	16384	32768
k-NN	67.8	68.5	70.0	71.2	71.8	71.9	70.6

larger memory benefits the learned representations up to a certain point where performance saturates. Based on these experiments, we set the memory size $K = 65536$ unless otherwise stated.

5.2. The Effect of the Memory Block Size

While the memory M controls the span to which the model can remember previous concepts, the memory block size N_b controls the dimensionality of the optimization problem. If N_b is too small, we might limit the variability of concepts to which we compare the current image views to representations from past iterations. If N_b is too large, we might encounter stability problems due to the weak self-supervised signal from the augmented views.

In Table 12, we investigate how the block size hyperparameter affects our framework. We report top-1 accuracy using k-NN for many configurations of N_b while keeping the memory size $K = 65536$. Empirically, MaSSL is robust to many configurations of N_b and does not collapse even when using very large block sizes. The experiments suggest an optimal value of $N_b = 16384$.

5.3. Sampling Memory Blocks

In Table 13, we compare different strategies to create memory blocks M_b from the main memory $\mathcal{M}$. We consider two protocols. **Stochastic:** A Memory block contains randomly sampled (without replacement) representations from the memory. **Blockwise:** A memory block is a contiguous section of representations from the main memory.

Empirically, the Blockwise approach for memory blocks collapses regardless of the block size N_b. This failure may be due to the FIFO update rule of the memory, which adds two properties to the learning mechanism. First, FIFO updates add an ordering/sequence bias in the location of the representations in memory, in which representations from one end of the memory are older than representations on the other end. Second, the FIFO updates shift (by a constant value) representations at each iteration towards the end of the memory. These update patterns make it easier for the network to *overfit* to its memory and collapse the representations.

Table 13. Strategies for sampling memory blocks. We report k-NN top-1 accuracy for varying block sizes N_b.

Block Strategy	512	1024	2048	4096
Stochastic	67.8	68.5	70.0	71.2
Blockwise	0.1	0.1	0.1	0.1

6. Discussion

Connection with SSL Clustering Methods. Self-supervised clustering methods (Caron et al., 2020; 2021; Silva & Ramírez Rivera, 2022; Silva & Ramírez Rivera, 2021) usually learn a set of prototypes using gradient descent. The biggest challenge in this setup is avoiding training collapse by solving the cluster assignment problem. Alternative approaches (Li et al., 2021) use classic machine learning algorithms such as k-means to bootstrap centroids and pose classification problems over the views. Regardless of the strategy, however, these methods usually require an explicit regularizer to avoid collapsed solutions. We can view MaSSL from a clustering perspective where a set of randomly chosen embeddings are selected at each iteration to act as anchors or centroids. Intuitively, MaSSL's learning process may be seen as a form of randomly bootstrapping centroids from memory, which acts as a form of approximation of the training data embedding manifold. Once initialized, these centroids are used to compute similarity scores across views. This perspective hints that the memory size K plays an important role and might depend on the number of hidden classes in the training dataset. Intuitively, the memory must be large enough to hold a fair number of examples from each class, increasing the probability of a good initialization of the prototypes, cf. Table 11.

7. Conclusion

We presented MaSSL, a memory-augmented self-supervised model for visual feature learning. MaSSL draws on the intuitive properties of memory to use information from past training iterations to learn invariant representations for the current image views. MaSSL offers interesting aspects such as (1) the use of its memory component in the SSL task, (2) the stochastic memory block sampling to regularize training, and (3) the lack of additional regularizers to avoid collapse. Moreover, MaSSL training architecture is simple and relatively cheaper to train. We provided many experimental results demonstrating our method's effectiveness in transfer and retrieval tasks.

Acknowledgements

The computations were performed in part on resources provided by Sigma2—the National Infrastructure for High Performance Computing and Data Storage in Norway—through

Project NN8104K. This work was funded in part by the Research Council of Norway, through its Centre for Research-based Innovation funding scheme (grant no. 309439), and Consortium Partners. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES)—Finance Code 001.

Impact Statement

"This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here."[2]

References

Asano, Y. M., Rupprecht, C., and Vedaldi, A. Self-labelling via simultaneous clustering and representation learning. In *Inter. Conf. Learn. Represent. (ICLR)*, 2019.

Assran, M., Caron, M., Misra, I., Bojanowski, P., Joulin, A., Ballas, N., and Rabbat, M. Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples. In *IEEE Inter. Conf. Comput. Vis. (ICCV)*, 2021.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. Deep clustering for unsupervised learning of visual features. In *European Conf. Comput. Vis. (ECCV)*, pp. 132–149, 2018.

Caron, M., Bojanowski, P., Mairal, J., and Joulin, A. Unsupervised pre-training of image features on non-curated data. In *IEEE Inter. Conf. Comput. Vis. (ICCV)*, pp. 2959–2968, 2019.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. In *Adv. Neural Inf. Process. Sys. (NeurIPS)*, 2020.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging Properties in Self-Supervised Vision Transformers. In *IEEE Inter. Conf. Comput. Vis. (ICCV)*, pp. 9650–9660, 2021.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *Inter. Conf. Mach. Learn. (ICML)*, 2020.

Chen, X. and He, K. Exploring simple siamese representation learning. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 15750–15758, 2021.

Chen, X., Xie, S., and He, K. An Empirical Study of Training Self-Supervised Vision Transformers. In *IEEE Inter. Conf. Comput. Vis. (ICCV)*, 2021.

Doersch, C., Gupta, A., and Efros, A. A. Unsupervised visual representation learning by context prediction. In *IEEE Inter. Conf. Comput. Vis. (ICCV)*, pp. 1422–1430, 2015.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., and Gelly, S. An image is worth 16x16 words: Transformers for image recognition at scale. In *Inter. Conf. Learn. Represent. (ICLR)*, 2020.

Douze, M., Jégou, H., Sandhawalia, H., Amsaleg, L., and Schmid, C. Evaluation of gist descriptors for web-scale image search. In *ACM Inter. Conf. on Image and Video Retr. (CIKM)*, pp. 1–8, 2009.

Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., and Zisserman, A. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *IEEE Inter. Conf. Comput. Vis. (ICCV)*, pp. 9588–9597, October 2021.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. In *Inter. Conf. Learn. Represent. (ICLR)*, 2018.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., and Azar, M. G. Bootstrap your own latent: A new approach to self-supervised learning. In *Adv. Neural Inf. Process. Sys. (NeurIPS)*, 2020.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 9729–9738, 2020.

Li, J., Zhou, P., Xiong, C., and Hoi, S. C. Prototypical contrastive learning of unsupervised representations. In *Inter. Conf. Learn. Represent. (ICLR)*, 2021.

Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., and Yu, S. X. Large-scale long-tailed recognition in an open world. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 2537–2546, 2019.

Loshchilov, I. and Hutter, F. Fixing weight decay regularization in adam, 2018. URL https://openreview.net/forum?id=rk6qdGgCZ.

Mairal, J. Cyanure: An open-source toolbox for empirical risk minimization for Python, C++, and soon more. *arXiv preprint arXiv:1912.08165*, 2019.

[2] Verbatim statement according to the Call for Papers guidelines, https://icml.cc/Conferences/2024/CallForPapers.

Misra, I. and Maaten, L. v. d. Self-supervised learning of pretext-invariant representations. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 6707–6717, 2020.

Noroozi, M. and Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conf. Comput. Vis. (ECCV)*, pp. 69–84. Springer, 2016.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. In *Wksp. Adv. Neural Inf. Process. Sys. (NeurIPSW)*, 2018.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.-Y., Xu, H., Sharma, V., Li, S.-W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., and Bojanowski, P. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2023.

Radenović, F., Iscen, A., Tolias, G., Avrithis, Y., and Chum, O. Revisiting Oxford and Paris: Large-scale image retrieval benchmarking. In *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 5706–5715, 2018.

Revaud, J., Almazán, J., Rezende, R. S., and Souza, C. R. d. Learning with average precision: Training image retrieval with a listwise loss. In *IEEE Inter. Conf. Comput. Vis. (ICCV)*, pp. 5107–5116, 2019.

Silva, T. and Ramírez Rivera, A. Representation learning via consistent assignment of views to clusters. In *IEEE Inter. Symp. Applied Comput. Intell. Inf. (SACI)*, pp. 987–994, 2022. ISBN 9781450387132. doi: 10.1145/3477314.3507267.

Silva, T. and Ramírez Rivera, A. Representation learning via consistent assignment of views over random partitions. In *Adv. Neural Inf. Process. Sys. (NeurIPS)*, 2023. URL https://openreview.net/forum?id=fem6BIJkdv.

Silva, T. S. and Ramírez Rivera, A. Consistent assignment for representation learning. In *Energy Based Models Wksp. (ICLRW)*, 2021.

Silva, T. S., Pedrini, H., and Ramírez Rivera, A. Self-supervised learning of contextualized local visual embeddings. In *IEEE Inter. Conf. Comput. Vis. Wksps. (ICCVW)*, 2023.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *Inter. Conf. Mach. Learn. (ICML)*, pp. 10347–10357. PMLR, 2021.

Xiao, K., Engstrom, L., Ilyas, A., and Madry, A. Noise or signal: The role of image backgrounds in object recognition. In *Inter. Conf. Learn. Represent. (ICLR)*, 2020.

Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., and Kong, T. iBOT: Image BERT Pre-Training with Online Tokenizer. In *Inter. Conf. Learn. Represent. (ICLR)*, 2022.

Table A.1. **Transfer learning evaluation**. We compare the top-1 k-NN accuracy of 9 SSL methods on 8 datasets. We report results for $k \in \{10, 20, 100, 200\}$.

		Oxford-IIIT Pet				Oxford Flowers-102				Aircraft				Stanford Cars			
Method	Arch	10	20	100	200	10	20	100	200	10	20	100	200	10	20	100	200
MoCo-v3	ViT-S/16	34.5	34.6	33.3	30.9	51.3	50.2	41.6	37.1	16.9	17.5	16.1	15.4	9.5	9.4	9.0	8.2
DINO	ViT-S/16	91.8	91.2	90.7	90.2	83.4	82.5	81.6	82.0	39.9	40.1	35.9	33.4	27.8	27.9	27.2	26.5
iBOT	ViT-S/16	91.8	91.4	90.8	90.5	83.4	82.0	80.9	81.2	39.7	39.3	36.0	32.8	25.8	25.5	24.7	23.2
Ours	ViT-S/16	91.5	91.9	90.4	90.5	84.5	83.6	82.6	83.1	38.6	37.9	35.2	32.5	30.5	31.0	29.9	29.1
MAE	ViT-B/16	21.0	19.4	15.7	14.3	18.7	16.9	10.2	9.2	9.5	9.7	8.3	6.6	6.3	6.0	4.8	4.8
MoCo-v3	ViT-B/16	83.4	83.8	81.4	80.4	74.9	70.2	64.3	66.0	28.5	27.4	23.1	21.6	23.4	22.4	21.3	20.0
DINO	ViT-B/16	90.4	90.1	88.4	88.3	85.7	84.6	83.9	84.4	38.4	38.5	34.5	32.2	32.3	32.7	31.1	30.0
iBOT	ViT-B/16	89.1	89.2	88.0	87.9	84.5	83.4	82.6	83.2	35.1	33.7	31.0	28.6	28.6	28.8	27.8	26.6
Ours	ViT-B/16	91.9	91.6	90.9	90.9	85.3	84.6	84.3	84.4	42.0	41.1	36.6	34.6	33.0	33.3	32.9	32.4

		Country-211				Food-101				STL-10				GTSRB			
Method	EP	10	20	100	200	10	20	100	200	10	20	100	200	10	20	100	200
MoCo-v3	ViT-S/16	7.6	7.5	7.2	6.6	28.6	31.1	34.3	34.1	66.8	67.1	64.4	62.3	38.2	38.4	37.7	36.8
DINO	ViT-S/16	15.0	15.1	15.6	15.7	69.1	69.3	67.8	66.5	98.4	98.4	98.2	98.2	60.6	61.1	61.4	60.6
iBOT	ViT-S/16	14.6	14.8	15.3	15.4	70.2	70.5	68.8	67.1	98.8	98.8	98.6	98.5	61.5	62.0	61.6	60.7
Ours	ViT-S/16	14.5	15.0	15.5	15.6	69.7	70.3	68.9	67.7	98.3	98.3	98.2	98.0	64.9	65.7	65.2	64.2
MAE	ViT-B/16	5.0	5.0	4.3	4.0	11.1	11.9	12.4	11.8	66.8	64.6	54.8	48.2	28.6	27.6	24.6	22.6
MoCo-v3	ViT-B/16	14.2	14.3	13.1	12.5	64.2	64.5	62.2	60.3	97.7	97.5	96.7	95.6	55.8	56.1	55.1	54.2
DINO	ViT-B/16	15.5	15.9	16.1	16.3	70.4	70.7	69.2	67.6	98.9	98.9	98.8	98.7	64.4	64.7	64.7	64.1
iBOT	ViT-B/16	15.4	15.7	16.1	16.1	72.0	72.6	70.4	68.8	99.0	99.0	98.9	98.8	62.9	63.0	61.4	60.3
Ours	ViT-B/16	15.4	15.7	16.2	16.2	72.1	72.5	71.1	69.7	98.7	98.8	98.6	98.5	68.4	69.3	69.0	67.9

A. Evaluation Protocols

A.1. Transfer Learning with k-NN and logistic regression Models

k-NN evaluation. We strictly follow the protocol and evaluation scripts from (Silva & Ramírez Rivera, 2023) for transfer learning using k-NN classifiers. We evaluate the following 8 datasets: Oxford-IIIT Pet, Oxford Flowers-102, AirCraft, Standard Cars, Country, Food-101, STL-10, and GTSRB. For all experiments, we run k-NN with configurations of $k \in \{10, 20, 100, 200\}$, and report the full results in Table A.1 where we compare the performance of different SSL methods using the ViT-S and -B backbones for all datasets across 4 values of k.

Logistic regression evaluation. In Table 2, we report the transfer learning performance by training logistic regression models on top of the frozen features of the pre-trained ViT-B encoder. We use the `cyanure` library (Mairal, 2019) logistic regression implementation and the same set of hyper-parameters for all models. Below, we show the pseudo-code used to create the logistic regression classifier object using `cyanure`.

```python
classifier = Classifier(loss="logistic", penalty="l2",
    solver="catalyst-miso", warm_start=False,
    max_iter=args.epochs,
    duality_gap_interval=10,
    fit_intercept=False,
    tol=1e-3,
    random_state=0,
    lambda_1=0.000002,
    lambda_2=0.000002)

classifier.fit(X, y)
```

A.2. Linear Probing and k-NN Evaluations on ImageNet

Linear probing on ImageNet-1M. We closely follow the protocol and code scripts from (Zhou et al., 2022) to train linear classifiers on the ImageNet-1M dataset on top of frozen features from pre-trained SSL methods. The evaluation script trains linear models with SGD, sweeps over different learning rates, and outputs the best model.

A.3. Low-Shot and Long-Tailed Evaluations

Low-shot on ImageNet.

Due to reproducibility issues, we adapted the linear probing evaluation script and reran the low-shot classification experiments for DINO, iBOT, and MoCo-v3 using available subsets for 1% and 10% ImageNet labeled images from Chen et al. (2020), cf. Table 6. Likewise, the evaluation script trains linear classifiers with SGD and sweeps over multiple learning rates. For low-shot evaluations using logistic regression on top of the frozen features, we use the `cyanure` library (Mairal, 2019).

We train linear models with SGD on balanced subsets of the ImageNet dataset, where we allow a fixed number of examples per class. In Table 7, we report top-1 accuracy for three versions of the ImageNet data where only one, two, and four images are randomly sampled per class. We repeat the experiments 5 times and report top-1 accuracy and standard deviations.

Long-tailed learning on ImageNet. To validate MaSSL representations on highly unbalanced data, we train linear models (single layer MLPs) on the ImageNet-LT dataset (Liu et al., 2019), which was designed as a long-tailed version of the ImageNet-2012. Its sampling strategy follows a Pareto distribution with a power value $\lambda = 6$. The ImageNet-LT contains 115.8K images with a maximum of 1280 and a minimum of five images per class. In Table 7, we report performance (top-1 accuracy) on the balanced ImageNet-LT test set.

B. Implementation Details

We train a joint-embedding teacher-student architecture using ViTs as backbones. We create multiple views of an image using different augmentation protocols. At each training iteration, we create 12 views from an image, 2 global views, each of size 224×224, and 10 local views, each of size 96×96. We follow the same augmentation protocol previously utilized by (Grill et al., 2020), namely a combination of color jittering, Gaussian blur, solarization, and random crop.

The student f and teacher g branches have different ViT encoders and projection heads. The projection head follows the same architecture proposed by (Caron et al., 2020), i.e., a multilayer perceptron (MLP) with 3 layers, hidden size is 2048-d, and Gaussian error linear units (GELU) activations. Only the student branch receives gradient updates. The teacher branch is updated following an exponential moving average from the student's network weights.

We only consider the `[CLS]` token from the Transformer encoder. For reference, the ViT-B encodes image views to representation vectors of 768-d, which are then projected to a lower 256-d and normalized to have a unit hypersphere.

The memory M is a non-differentiable container that holds representations at each training iteration and is updated following a FIFO (First-In, First-Out) strategy. The memory size is set to $K = 65536$ following the ablations experiments in Section 5.1. Before optimization, the view-memory similarity distribution is split into disjoint subsets called memory blocks, each of size $N_b = 16384$, Cf. Section 3.2.

MaSSL is trained with the AdamW optimizer (Loshchilov & Hutter, 2018), learning rate 1×10^{-5}, and a global batch size of 1024. The learning rate follows a cosine decay without warmup towards 1×10^{-6}. Following (Caron et al., 2021), the weight decay follows a cosine schedule from 0.04 to 0.4. The student temperature is set to $\tau_s = 0.1$, and the teacher temperature τ_t is warmed up from 0.04 to 0.07 in the first 30 epochs.

B.1. PyTorch Style Pseudo-code

```
# D: Images' representation dimensionality
# K: Memory size
# NB: Memory block size
# B: Number of memory blocks
# N: Batch size
# z_i: Representation vector from the student encoder
# w_i: Representation vector from the teacher encoder

memory = torch.randn(D, K)
memory = F.normalize(memory, dim=0)

for x1, x2 in loader:
    # student and teacher branches
    z1, w1 = f(x1), g(x1) # [N, D]
    z2, w2 = f(x2), g(x2) # [N, D]
    p1, p2 = matmul(z1, memory), matmul(z2, memory) # [N, K]
    q1, q2 = matmul(w1, memory), matmul(w2, memory) # [N, K]

    # sample cluster indices with no replacement
    rand_proto_ids = torch.randperm(K)
    split_embed_ids = stack(split(rand_proto_ids, NB))

    ps, qs = [], []
    for p, q in zip([p1, p2], [q1, q2]):
        p_mb = fetch_mem_block(p, split_embed_ids)
        q_mb = fetch_mem_block(q, split_embed_ids)

        ps.append(p_mb)
        qs.append(q_mb)

    ps, qs = torch.cat(ps, dim=0), torch.cat(qs, dim=0)
    loss = loss_fn(ps, qs)

    # update memory
    enqueue(memory, w1)
    dequeue(memory)

    # gradient descent steps


def loss_fn(ps, qs):
    for i in range(len(ps)):
        for j in range(len(qs)):
            if i == j:
                continue
            consistency += cross_entropy(ps[i], qs[j])
            terms += 1
    consistency /= terms
    return consistency

def cross_entropy(p, q):
    p = torch.log_softmax(p, dim=-1)
    q = torch.softmax(q, dim=-1)

    loss = torch.sum(-q * p, dim=-1)
    return loss

def fetch_mem_block(logits, proto_ids):
    logits_gr = logits[:, proto_ids.flatten()]
    logits_gr = logits_gr.split(NB, dim=1)
    logits_gr = torch.cat(logits_gr, dim=0)
    return logits_gr # [N * B, NB]
```

C. Additional Results

C.1. Visualizing Self-Attention Maps

We provide additional self-attention visualizations in Figure C.1. We strictly follow the generating scripts from Zhou et al. (2022), and display attention maps from pre-trained ViT-S backbones using images sampled from the validation set of the ImageNet-1M dataset, hence not used for training. In Figure C.1, for each image, we show attention maps from MaSSL, iBOT, and DINO in this order from top to bottom. The protocol uses the [CLS] token as a query to extract attention maps over multiple heads of the last layer. MaSSL learns comparable attentive maps to iBOT and DINO, where we can see the attention maps segmenting the object in the image and different heads paying attention to different features in the image.

C.2. Sparse Correspondence

We follow the sparse correspondence evaluation protocol proposed by (Zhou et al., 2022), where the task is to match patch embeddings from the last layer of the ViT. We qualitatively compare MaSSL against iBOT and DINO using ViT-S/16 backbones pre-trained for 800. We consider two cases: (1) the pair of matching images contain views extracted from the same image (Figure C.2) and (2) the pair contains two distinct images from the same class (Figure C.3). The protocol matches local embeddings from two images, and at most 14×14 matched pairs can be extracted with a ViT-S. The evaluation script displays the 12 correspondences with the highest self-attention scores. In Figure C.3, we show examples of feature correspondences for image pairs drawn from a wide variety of classes containing buildings, animals, humans, vehicles, and other objects. MaSSL can extract mostly correct correspondences despite augmentations on scale and color.

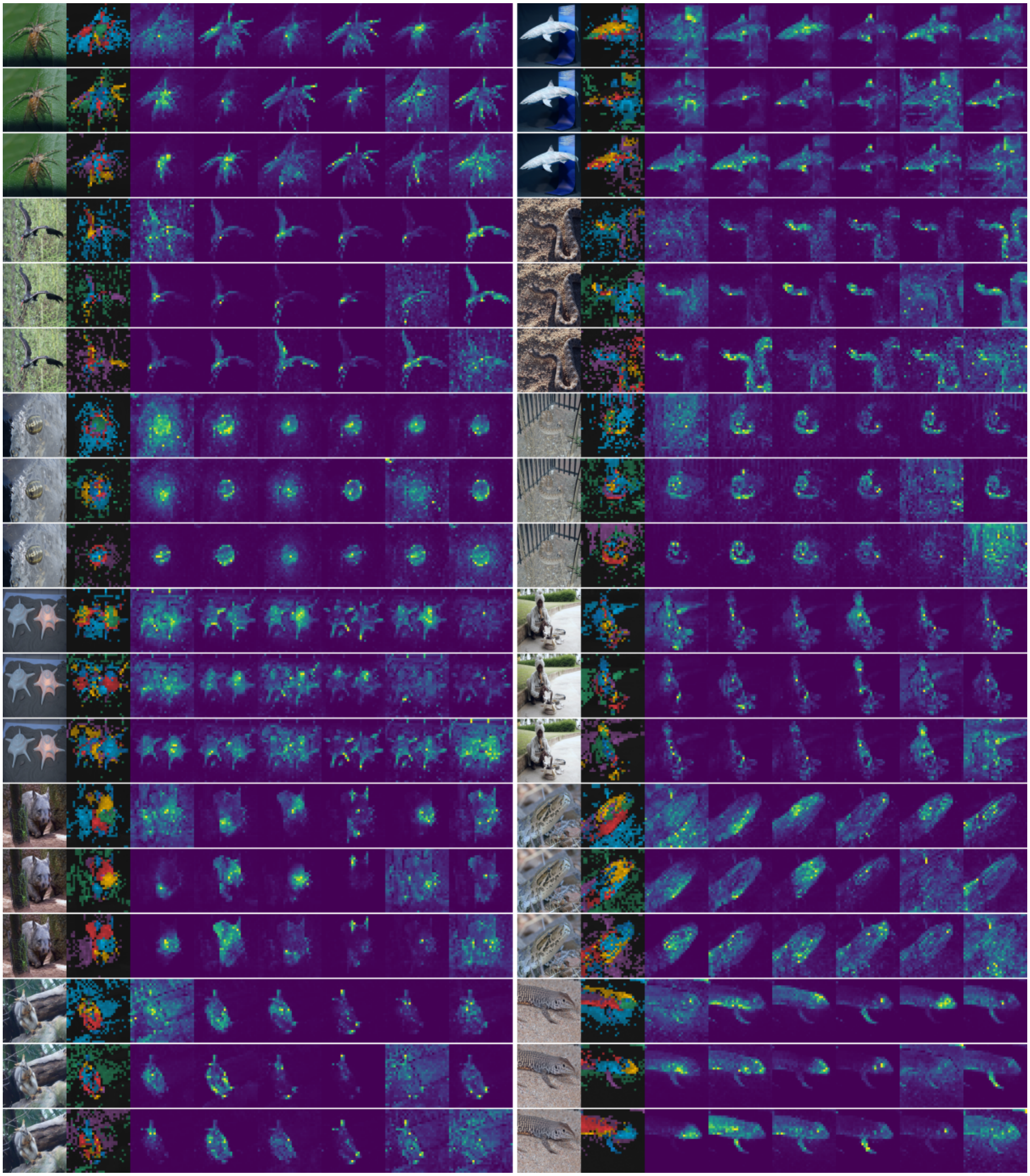

Figure C.1. **Visualizing self-attention maps.** From top to bottom, in each triplet of rows, we report qualitative evaluations for MaSSL, iBOT, and DINO. The columns show multiple attention heads of the last layer.

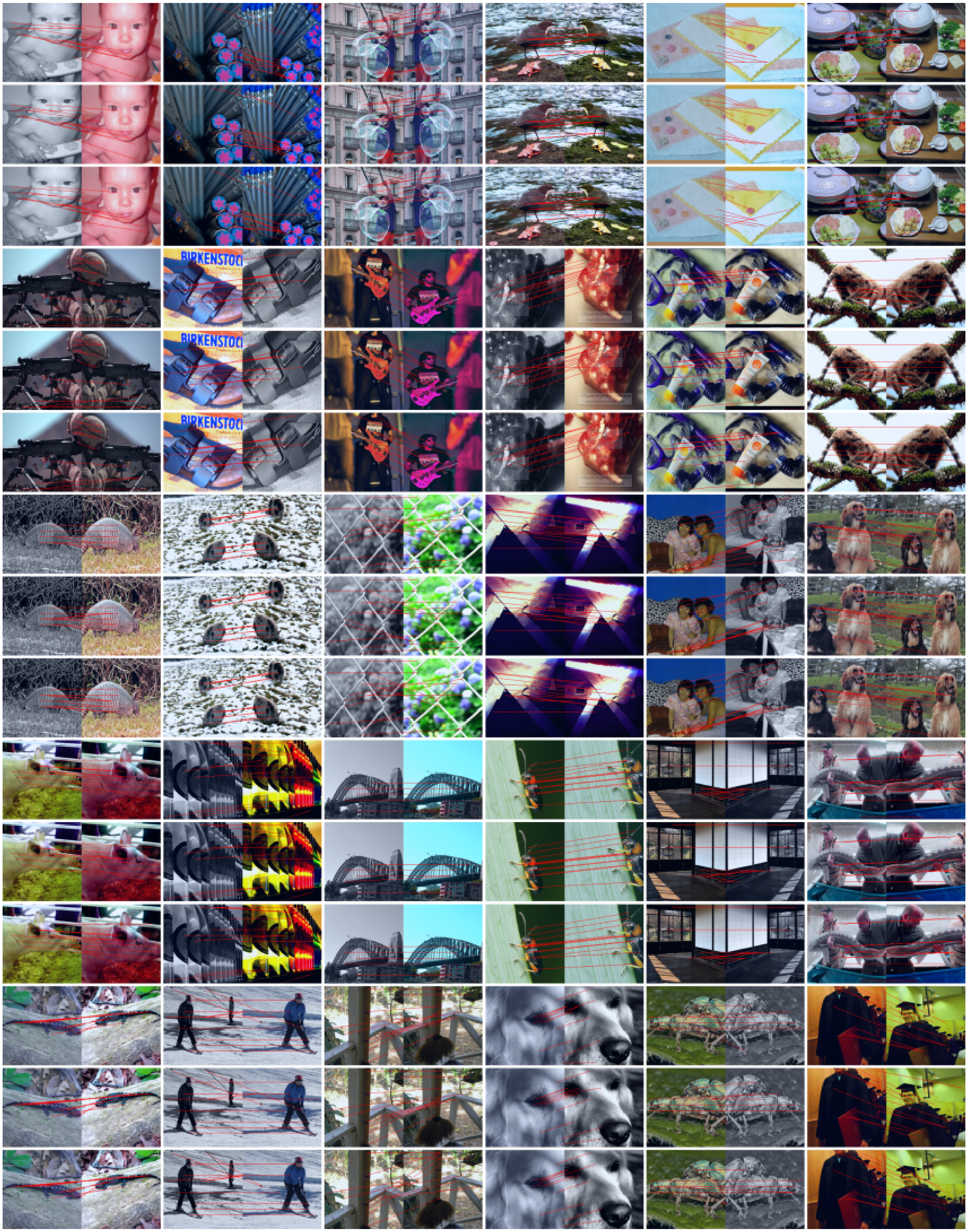

Figure C.2. **Visualization for sparse correspondence.** We assess the ability to match local embeddings using pairs of views from the same image. From top to bottom, in each triplet of rows, we report qualitative evaluations for MaSSL, iBOT, and DINO.

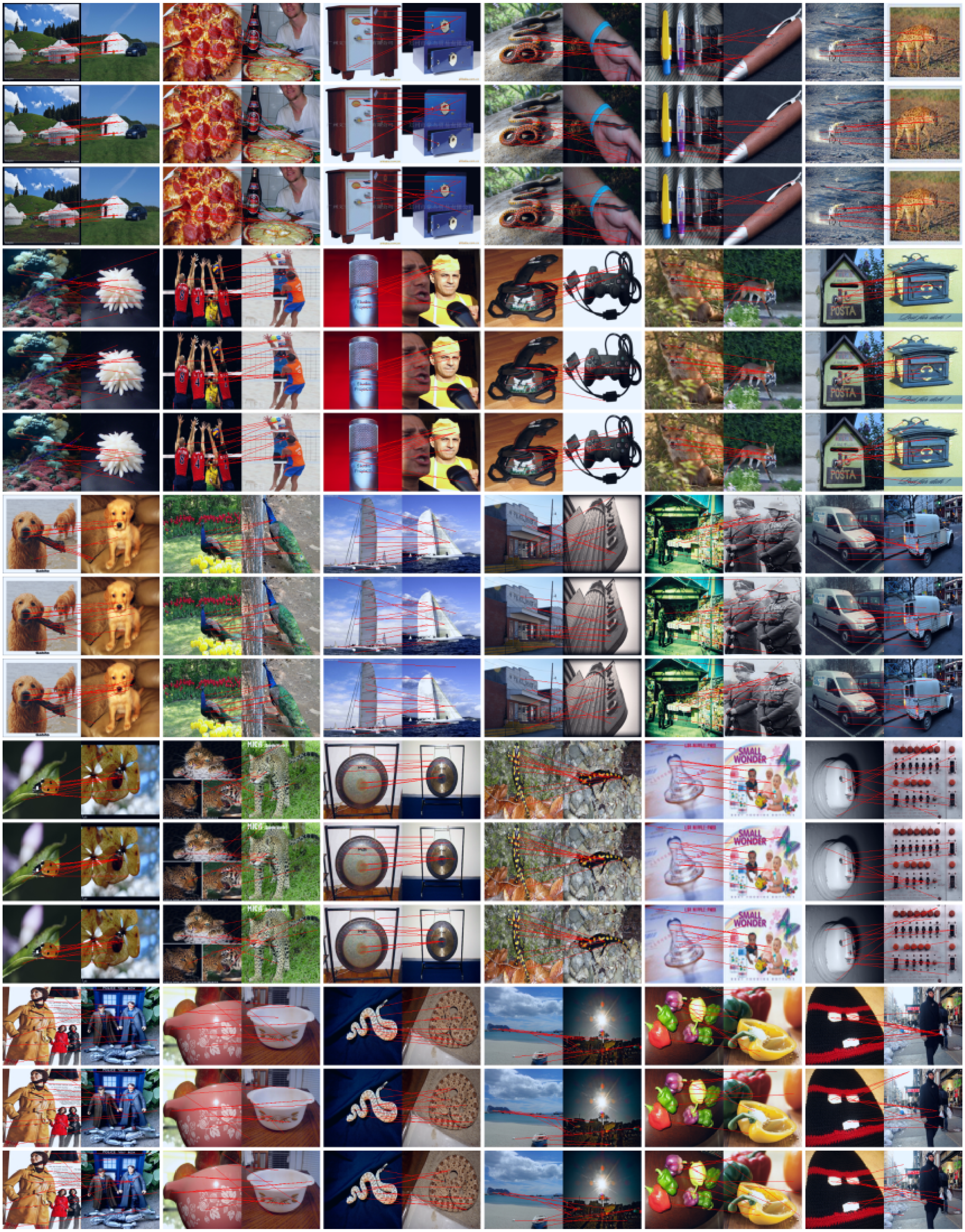

Figure C.3. **Visualization for sparse correspondence.** We assess the ability to match local embeddings using a pair of images from the same class. From top to bottom, in each triplet of rows, we report qualitative evaluations for MaSSL, iBOT, and DINO.