FLSL: Feature-level Self-supervised Learning

Qing Su[1], **Anton Netchaev**[2], **Hai Li**[3], **and Shihao Ji**[1]
[1]Georgia State University, [2]U.S. Army ERDC, [3]Duke University

Abstract

Current self-supervised learning (SSL) methods (*e.g.*, SimCLR, DINO, VICReg, MOCOv3) target primarily on representations at instance level and do not generalize well to dense prediction tasks, such as object detection and segmentation. Towards aligning SSL with dense predictions, this paper demonstrates for the first time the underlying *mean-shift* clustering process of Vision Transformers (ViT), which aligns well with natural image semantics (*e.g.*, a world of objects and stuffs). By employing transformer for joint embedding and clustering, we propose a two-level feature clustering SSL method, coined Feature-Level Self-supervised Learning (FLSL). We present the formal definition of the FLSL problem and construct the objectives from the *mean-shift* and k-means perspectives. We show that FLSL promotes remarkable semantic cluster representations and learns an embedding scheme amenable to *intra-view* and *inter-view* feature clustering. Experiments show that FLSL yields significant improvements in dense prediction tasks, achieving 44.9 (+2.8)% AP and 46.5% AP in object detection, as well as 40.8 (+2.3)% AP and 42.1% AP in instance segmentation on MS-COCO, using Mask R-CNN with ViT-S/16 and ViT-S/8 as backbone, respectively. FLSL consistently outperforms existing SSL methods across additional benchmarks, including UAV object detection on UAVDT, and video instance segmentation on DAVIS 2017. We conclude by presenting visualization and various ablation studies to better understand the success of FLSL.

1 Introduction

Following its success in natural language processing (NLP) [40, 5, 17], self-supervised learning (SSL) with transformer [49, 19] has emerged as a highly effective strategy and a popular model choice over the CNN-based counterparts in vision tasks. The remarkable performance achieved by SSL have been demonstrated by SimCLR [11], MOCOv3 [13], DINO [8], VICReg [3], SwAV [7], BYOL [23], and among others. Without relying on manual supervision, a successful paradigm of SSL promotes semantic representations conducive to the downstream tasks, *e.g.*, classification, detection and segmentation. However, most existing SSL methods operate at the instance-level, where an encoder is trained to maximize the agreement of the representations of multiple augmented views of an image. Though demonstrating strong performance on the classification tasks [11, 25], the instance-level SSL is inherently misaligned with the dense prediction tasks, such as object detection, where the lower level semantic information plays a bigger role than the instance-level semantic information. This leads to inferior transferability to those dense prediction tasks.

Recent attempts to bridge the semantic gap are mainly based on region [43], patch [55, 18], or pixel (*i.e.*, dense feature) matching tasks [51, 58, 32] with optional instance-level objectives. However, learning of distinct representation for each image patch or region still mismatches the natural semantics within an image (referred to as local semantics), where features of the same semantics should be highly correlated other than being distinct. Semantics can range from features of high similarity, features of the same object, to more complex semantic structures. Methods such as SoCo [52] and ORL [56] leverage the off-the-shelf *selective search* [47] to impose the semantic constraint to the contrastive learning pipeline. Nonetheless, the inclusion of a non-trainable region

Preprint. Under review.

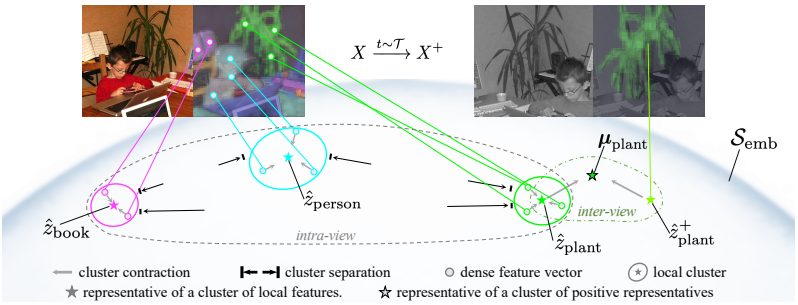


Figure 1: The two-level clustering of FLSL. An object or stuff in an image is essentially a cluster of features. Their representations can be extracted as cluster representatives, *e.g.*, modes. FLSL promotes the representations to be simultaneously ***locally*** and ***globally*** semantic. On the embedding sphere $\mathcal{S}_{emb}$, the feature clusters of objects or stuff (book, person, plant, etc.) in an image X are learnt to be compact and well-separated, *i.e.*, features are driven close to their cluster representatives $\hat{z}_{\text{cls}}$s and far away from features of other clusters (*intra-view* clustering). Meanwhile, cluster representatives $\hat{z}_{\text{cls}}$s are clustered by pushing them closer to their positive samples $\hat{z}^+_{\text{cls}}$s in X^+, augmented via the transformation $t \sim \mathcal{T}$ (*inter-view clustering*), such that they encode the same category information and become ***globally*** semantic.

proposal module in both methods impedes the learning of distinct representations of RoIs among each other and from the rest of the image, which is the desired property of locally semantic representations for object detection.

Existing SSL methods targeting dense prediction primarily focus on learning globally semantic representations of image sub-regions as RoIs, patches, or pixels with limited consideration for the alignment of those representations with local semantics. This observation leads us to ask the following question: Can we learn a representation that is both locally and globally semantic for a group of features (*e.g.*, representing an object) in an end-to-end trainable SSL approach? To this end, we propose the *Feature Level Self-supervised Learning* (FLSL) that leverages the *mean-shift* clustering process inherent in the transformer to extract the representatives of feature clusters as representations and incorporates *k-means* based SSL approach to induce the learned representations both locally and globally semantic. Figure. 1 illustrates the main idea of FLSL with details to be discussed in Sec. 4.

Contributions This paper takes a step forward to bridge the gap between the current SSL methods and downstream dense prediction tasks. Our contributions are summarized as follows:

1. We demonstrate for the first time the connection between the attention mechanism and *mean-shift* clustering, and reinterpret vision transformer from the perspective of *mean-shift*.
2. By employing transformer for joint embedding and feature clustering, we propose FLSL, an end-to-end trainable SSL method that promotes the representations of feature clusters to be semantic at two levels: (i) intra-view clusters within an image, and (ii) inter-view clusters over an entire dataset.
3. The derivation and construction of the FLSL objectves is rooted in *mean-shift* and the non-empty *k-means* clustering. The first-level semantic representation is encouraged by optimizing the intra-cluster feature affinity with a self-attention layer, while the second-level semantic representation is encouraged through the non-empty *k-means* clustering with positive samples retrieved through a cross-attention layer.
4. We validate the synergy between FLSL and ViT, and show significant improvement in transferability of learnt features to dense prediction tasks, including object detection and segmentation. FLSL-pretrained ViT on ImageNet-1k (IN1k) demonstrates superior performance compared to the state-of-the-art ADCLR-IN1k [61] and MAE [33] pretrained counterparts. Moreover, it consistently outperforms existing SSL methods across additional benchmarks, including UAV object detection on UAVDT, and video instance segmentation on DAVIS 2017.

2 Related work

SSL for dense prediction Recent attempts to bridge the gap between common SSL and dense prediction tasks focus primarily on sub-region matching tricks. For example, DenseCL [51] applies contrastive learning on pairs of patches with highest similarity. However, the patch-matching trick leads to distinct representations with low correlation among patches, which is ill-posed for the semantics of a natural image. Along with the instance-level objective, PixPro [58] and LC-loss [29] factor in agreement between positive pixel pairs which are assigned through thresholded-distance in PixPro and position projection in LC-loss. DetCo [55] further incorporates instance-patch level

contrastive losses along with instance level and patch level losses. To learn representations at object level, SoCo [52] and ORL [56] employ *selective search* to crop out RoIs. ORL further enables inter-object contrastive learning via top-ranked RoI pair retrieval. In contrast, SCRL [43] relaxes the semantic constraint using random crops within the intersection area of augmented views as RoIs. As discussed in Sec. 1, all of these methods focus on learning globally semantic representations for image sub-regions, and they do not touch on local semantics that are necessary for dense prediction.

Self-supervised vision transformer In pioneering works, self-supervised training of transformer for vision tasks generally follow the paradigm of masked autoencoder in NLP [40, 17]. For instance, iGPT [10] features reconstruction of masked pixels as one of its objectives. In general, SSL for ViT can be classified into two categories: the joint-embedding strategy epitomized by DINO [8] and MoCov3 [13], and the generative approaches represented by MAE [24]. The crossover of the two strategies is demonstrated by iBOT [62]. Regarding **dense prediction**, EsViT [32], designed for Swin Transformer [36], follows the region-matching strategy and applies the DINO loss to the probabilities of positive pairs determined by highest similarity. Instead of finding the best-matching patch, SelfPatch [60] considers the direct neighbors as its positive patches. However, with limited semantics contained in a fixed small area (*e.g.*, 8-connected neighbors), the method still suffers from semantic misalignment. To address the sub-region mismatch issue of DINO, ADCLR [61] constructs query tokens from random sub-regions and treats them as extra class tokens in the DINO objective. This promotes region-aware semantic representations that better aligned with the local semantics, and leads to substantial improvement in dense prediction.

3 Intuition: the connection between mean-shift and attention

As discussed in Sec. 1, the misalignment between the current SSL methods and dense prediction tasks lies in the clustering bias at the semantic level. Instead of setting a fixed granularity, such as instance-level or fix-sized patch-level, a desired semantic representation scheme should be able to represent from a single patch to a cluster of patches or even an entire image. The representation space of an image can be considered as an empirical probability density function of features, and the modes (local maxima) therefore can be regarded as the representatives of clusters [9, 14, 15]. These modes can then be readily retrieved via clustering algorithms, particularly, non-parametric *kernel density estimation* (KDE) methods [50] when the image composition (*e.g.*, number of objects and stuffs) is unknown. One typical KDE-based method is the *mean-shift* clustering [28]. In the following, we first give an overview of self-attention (SA) mechanism of transformer and the mean-shift algorithm. We then show that the mean-shift update rule conforms to the SA mechanism of transformer.

Attention mechanism First introduced to recurrent neural networks as a context extractor for machine translation [2], attention has premised major breakthroughs in NLP with the emergence of transformer that relies solely on the *scaled dot-product* attention mechanism [49] given by

$$\text{attention}\left(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}\right) = \boldsymbol{V}\text{softmax}\Big(\boldsymbol{Q}^\top \boldsymbol{K}/\sqrt{D_{qk}}\Big), \tag{1}$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ denote query, key and value matrices packing together sets of query, key and value vectors, respectively, D_{qk} denotes the dimension of query and key vectors, and $\text{softmax}\left(\boldsymbol{Z}\right)_{ij} = \exp(\boldsymbol{Z}_{ij})/\sum_k \exp(\boldsymbol{Z}_{ik})$. As a special case of attention, SA matches a sequence $\boldsymbol{Z}$ with itself to extract the semantic dependencies among its components, *i.e.*, $\boldsymbol{Q} = \mathbf{W}_Q \boldsymbol{Z}, \boldsymbol{K} = \mathbf{W}_K \boldsymbol{Z}, \boldsymbol{V} = \mathbf{W}_V \boldsymbol{Z}$, where the projections $\mathbf{W}\_$'s are the parameter matrices.

Mean-shift clustering and attention Given N data points $\{\boldsymbol{z}_i\}_{i=1}^N \subset \mathbb{R}^D$, the kernel density estimate of $p(\boldsymbol{z})$ with kernel $K(t)$ can be defined as

$$p(\boldsymbol{z}) = \sum_{i=1}^{N} p(\boldsymbol{z}_i) p(\boldsymbol{z}|\boldsymbol{z}_i) = \sum_{i=1}^{N} \pi_i \frac{1}{T_i} K(d(\boldsymbol{z}, \boldsymbol{z}_i; \boldsymbol{\Sigma}_i)), \tag{2}$$

where $p(\boldsymbol{z}_i) = \pi_i$ is the mixing proportion of point $\boldsymbol{z}_i$, *s.t.* $\sum_{i=1}^N \pi_i = 1$, T_i denotes the normalization term dependent only on the covariance matrix $\boldsymbol{\Sigma}_i$, *e.g.*, for a Gaussian kernel $T_i = |2\pi\boldsymbol{\Sigma}_i|^{1/2}$ and $d(\boldsymbol{z}, \boldsymbol{z}_i; \boldsymbol{\Sigma}_i) = (\boldsymbol{z} - \boldsymbol{z}_i)^T \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{z} - \boldsymbol{z}_i)$ is the *Mahalanobis* distance. Finding the modes of $p(\boldsymbol{z})$ is to seek stationary points by equating the gradient of $p(\boldsymbol{z})$ to zero, $\partial p(\boldsymbol{z})/\partial \boldsymbol{z} = 0$, which arrives at

$$\hat{\boldsymbol{z}} = \boldsymbol{f}\left(\boldsymbol{z}\right) = \sum_{i=1}^{N} p(\boldsymbol{z}_i|\boldsymbol{z}) \boldsymbol{z}_i, \quad \text{with } \; p(\boldsymbol{z}_i|\boldsymbol{z}) = \frac{\pi_i \frac{1}{T_i} K'(d(\boldsymbol{z}, \boldsymbol{z}_i; \boldsymbol{\Sigma}_i)) \boldsymbol{\Sigma}_i^{-1}}{\sum_{j=1}^{N} \pi_j \frac{1}{T_j} K'(d(\boldsymbol{z}, \boldsymbol{z}_j; \boldsymbol{\Sigma}_j)) \boldsymbol{\Sigma}_j^{-1}}, \tag{3}$$

where $K' = dK/dt$. The above fixed-point iterative scheme is the *mean-shift* algorithm. Practically, on ℓ_2-normalized vectors, for a homoscedastic *Gaussian* kernel with constant mixing proportion and isotropic covariances (*e.g.*, $\pi_i = 1/N$, $1/\sigma^2 = \tau$), Eq. 3 further simplifies to

$$\hat{\boldsymbol{z}} = \text{meanshift}(\boldsymbol{z}, \tau) = \sum_{i=1}^{N} \frac{\exp\left(\tau \boldsymbol{z}^\top \boldsymbol{z}_i\right)}{\sum_{j=1}^{N} \exp\left(\tau \boldsymbol{z}^\top \boldsymbol{z}_j\right)} \boldsymbol{z}_i \Longrightarrow \hat{\boldsymbol{Z}} = \boldsymbol{Z}\, \text{softmax}\left(\tau \boldsymbol{Z}^\top \boldsymbol{Z}\right), \tag{4}$$

which conforms to the attention function (Eq. 1) with identity projection matrices, *i.e.*, $\boldsymbol{W}_Q = \boldsymbol{W}_K = \boldsymbol{W}_V = \mathbf{I}$, and $\tau = 1/\sqrt{D_{qk}}$. Conversely, the conventional SA mechanism can be viewed as a generalized *mean-shift*:

$$\hat{\boldsymbol{Z}} = \text{SA}(\boldsymbol{Z}) = \mathbf{W}_V\, \boldsymbol{Z}\, \text{softmax}\left({}^{1}/_{\sqrt{D_{qk}}} \boldsymbol{Z}^\top \left(\mathbf{W}_Q^\top \mathbf{W}_K\right) \boldsymbol{Z}\right), \tag{5}$$

with learnable distance measure $\boldsymbol{Z}^\top (\mathbf{W}_Q^\top \mathbf{W}_K) \boldsymbol{Z}$ and projection $\mathbf{W}_V$. Unlike GMM and *k-means*, *mean-shift* is capable of modeling clusters of complex non-convex shape with cluster number automatically determined by local scale (proscribed by covariance) [28]. Hence, it is well-aligned with the semantics of natural images.

ViT from the perspective of mean-shift In ViT [19], images are initially tokenized and then processed through a sequence of transformer layers. Each transformer layer is comprised of a skip-connected multi-head SA (MHSA) and a skip-connected MLP. MHSA can be constructed from Eq. 5 with m projections in parallel, *i.e.*, $[\mathbf{W}_Q^h, \mathbf{W}_K^h, \mathbf{W}_V^h], h = 1, \cdots, m$. The m returned modes are then concatenated along channel dimension and reprojected to a single return through

$$\hat{\boldsymbol{Z}} = \text{MHSA}(\boldsymbol{Z}) = \mathbf{W}_O \text{concat}([[\hat{\boldsymbol{Z}}^1], \ldots, [\hat{\boldsymbol{Z}}^m]]) + \mathbf{b}_O. \tag{6}$$

Note that the ℓ_2 normalization assumed in Eq. 4 is moderately relaxed through layer normalization (LN) to incorporate the extra degree of freedom in the vector magnitude. With skip connection and the one-step *mean-shift* update described in Eqs. 5, 6, a transformer layer essentially finds the local centroid of each query $\boldsymbol{z}$ and drives them closer to the (projected) local centroids through $\boldsymbol{z} = \hat{\boldsymbol{z}} + \boldsymbol{z}$, followed by an MLP processing step with skip connection. ViT iterates the process multiple times (*e.g.*, 12 or 24 layers) to capture the contextual and semantic information of an image.

The clustering process above concords with one inductive bias of the attention mechanism represented by the *sparse variable creation* [21], *i.e.*, an SA head learns a sparse function that only depends on a small subset of input coordinates. In the context of clustering, the subset of input corresponds to the modes of density $p(\boldsymbol{z})$ of features. As the high-level semantic information is typically spatially sparse (*e.g.*, the feature vector for a RoI in object detection, a single label for a segment in segmentation, or a scene-graph, etc.), it is natural to leverage transformer for joint embedding and clustering to learn semantic representations at the feature level.

4 Methodology

FLSL features a two-level clustering process (Figure 1), which is formally described as follows.

Given a dataset $\mathcal{X}$ (e.g., a set of images), FLSL learns an embedding scheme $f_\theta : \mathcal{X} \to \mathcal{Z}$, $\forall \boldsymbol{X} \in \mathcal{X}$, $\boldsymbol{Z} = f_\theta(\boldsymbol{X})$. $\boldsymbol{Z}$ can be formulated as $\boldsymbol{Z} = \bigcup_c^{N_c} \tilde{\boldsymbol{z}}^c$, where $\tilde{\boldsymbol{z}}^c$ is a subset of $\boldsymbol{Z}$ forming a cluster, N_c is the number of clusters determined by a clustering scheme, *e.g.*, *mean-shift*, and $N_c \leq |\boldsymbol{Z}|$. FLSL aims to encourage the following properties:
(i) **Intra-view**: embeddings within a cluster, $\boldsymbol{z} \in \tilde{\boldsymbol{z}}^c$, are close to the cluster representative (mode) $\hat{\boldsymbol{z}}^c$ and far away from the embeddings of other clusters;
(ii) **Inter-view**: the cluster representatives (modes) $\hat{\boldsymbol{z}}$s of the positive regions in $\boldsymbol{X}$s over $\mathcal{X}$ are pushed closer to each other.

The FLSL-extracted features should be well-aligned with dense prediction tasks, such as object detection, where the representation of an object or stuff (*i.e.*, cluster of features) are desired to be (i) well-separated from others in an image (locally semantic), and (ii) close to its positive samples in the dataset (globally semantic). In this section, we present the objectives for both levels of clustering, which are then combined to form the final objective.

4.1 Intra-view clustering with mean-shift

As discussed in Sec. 3, local semantics of an image can be captured by non-parametric clustering such as *mean-shift*. Hence, with *mean-shift* update rule Eq. 4, it can be proved that the posterior of $\boldsymbol{z}_j$

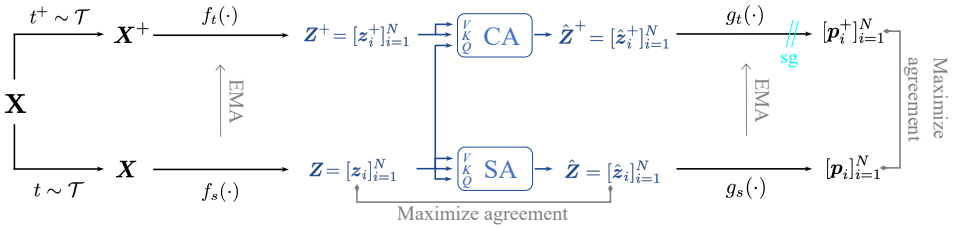


Figure 2: Overview of the FLSL framework. Similar to DINO [8], FLSL is comprised of a teacher network and a student network, which have the same architecture – a ViT encoder f and a projection head g – but with different parameters. Two *mean-shift* operations: a non-parametric self-attention (SA) and a non-parametric cross-attention (CA) are applied to the last layer of f_t, f_s before g_t, g_s, respectively, and the CA takes output of f_s as queries. The two networks are trained to maximize the agreement between the probability distributions $\boldsymbol{p}_i$s and $\boldsymbol{p}_i^+$s and the agreement between features $\boldsymbol{z}_i$s and their cluster representatives $\hat{\boldsymbol{z}}_i$s.

given point $\boldsymbol{z}_i$, $p(\boldsymbol{z}_j|\boldsymbol{z}_i) = [\text{softmax}(\tau \boldsymbol{z}_i^\top \boldsymbol{Z})]_j$, should satisfy:

$$p(\boldsymbol{z}_j|\boldsymbol{z}_i) \geq 1/\left(\left(\sum_{k\in c_i} e^{(\boldsymbol{z}_i^\top \boldsymbol{z}_k - \boldsymbol{z}_i^\top \boldsymbol{z}_j)\tau}\right) + (N - |c_i|)e^{-\Delta_{ij}\tau}\right), \forall j \in c_i \tag{7}$$

where $N = |\boldsymbol{Z}|$, c_i is the set of indices of points in the same cluster that point $\boldsymbol{z}_i$ belongs to, and Δ_{ij} is the degree of separability defined as $\Delta_{ij} = \boldsymbol{z}_i^\top \boldsymbol{z}_j - \max_{k\in[N]\setminus c_i} \boldsymbol{z}_i^\top \boldsymbol{z}_k$, such that larger Δ_{ij} indicates better separation. For locally semantic embeddings, we desire the in-cluster points to be close to each other, or equivalently, to be close to its cluster representative, and stay far away from the out-cluster points, which indicates a large Δ value. As Δ becomes sufficiently large, the RHS of Eq. 7 can be approximated as $1/\sum_{k\in c_i} \exp\left((\boldsymbol{z}_i^\top \boldsymbol{z}_k - \boldsymbol{z}_i^\top \boldsymbol{z}_j)\tau\right)$, and for out-cluster points, the posterior $p(\boldsymbol{z}_{j\notin c_i}|\boldsymbol{z}_i)$ approaches to 0. This results in a semantics-aligned cluster representative via *mean-shift* – a weighted sum of **only** in-cluster points. Assuming the out-cluster points are fixed, we can promote the above property by simply driving the returned mode $\hat{\boldsymbol{z}}_i$ and query point $\boldsymbol{z}_i$ closer to each other, which leads to the **intra-view** clustering objective:

$$\min_{f_\theta} \sum_{i=1}^{N} \|\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i\|_2^2. \tag{8}$$

Proof of Eq. 7 and detailed explanation is provided in Appendix A.

4.2 Inter-view clustering with k-means

To learn globally semantic representations, similar to the existing SSL methods, we formulate the problem as a variant of *k-means* clustering. In the space of cluster representatives $\hat{\boldsymbol{z}}$s extracted from an entire dataset, the *k-means* objective with generalized non-empty cluster constraint [4] can be expressed as

$$\min_{\mathcal{M}} \frac{1}{N'} \sum_{\hat{\boldsymbol{z}}\in\hat{\mathcal{Z}}} \sum_{k=1}^{K} \delta_{k k(\hat{\boldsymbol{z}})} \|\hat{\boldsymbol{z}} - \boldsymbol{\mu}_{k(\hat{\boldsymbol{z}})}\|_2^2 + D_{\text{KL}}\left(\bar{\boldsymbol{p}} \| \boldsymbol{\pi}\right), \tag{9}$$

where $\mathcal{M}$ is a set of K centroids $\{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K\}$, $\hat{\mathcal{Z}}$ is a set of cluster representatives over the entire dataset, $N' = |\hat{\mathcal{Z}}|$, $k(\hat{\boldsymbol{z}}) = \arg\min_k \|\boldsymbol{\mu}_k - \hat{\boldsymbol{z}}\|_2$, δ_{ij} is the *Kronecker delta*, with $\delta_{ij} = 1$ iff $i = j$, and 0 otherwise, $[\bar{\boldsymbol{p}}]_{[i]} = 1/N' \sum_{\hat{\boldsymbol{z}}} \delta_{ik(\hat{\boldsymbol{z}})}$, and $\boldsymbol{\pi}$ is the prior, *e.g.*, a vector of the preset proportion for each cluster. With positive pairs $(\hat{\boldsymbol{z}}^+, \hat{\boldsymbol{z}})$ via data augmentation, the objective can then be constructed as *k-means* clustering with an extra separation margin for $\hat{\boldsymbol{z}}^+$:

$$\min_{\mathcal{M}} \frac{1}{N'} \sum_{\hat{\boldsymbol{z}}\in\hat{\mathcal{Z}}} \left(\sum_{k=1}^{K} \delta_{k k(\hat{\boldsymbol{z}})} \|\hat{\boldsymbol{z}} - \boldsymbol{\mu}_{k(\hat{\boldsymbol{z}})}\|_2^2 + \left(1 - \delta_{k(\hat{\boldsymbol{z}}^+)k(\hat{\boldsymbol{z}})}\right) \|\hat{\boldsymbol{z}}^+ - \boldsymbol{\mu}_{k(\hat{\boldsymbol{z}})}\|_2^2 \right) + D_{\text{KL}}\left(\bar{\boldsymbol{p}} \| \boldsymbol{\pi}\right). \tag{10}$$

A common approach to tackle the optimization problem above is to relax the hard cluster assignment constraint $\delta_{ij} \in \{0, 1\}$ to $[0, 1]$ via a **classification head** with a small temperature ($\ll 1$) to $\hat{\boldsymbol{z}}$. This relaxes Eq. 9 to a more general Gaussian Mixture Model (GMM) formulation (cf. Appendix B).

By rewriting $1 - \delta_{k(\boldsymbol{z}^+)k(\boldsymbol{z})}$ in Eq. 10 as $\sum_{k=1}^{K} \delta_{k k(\boldsymbol{z}^+)} - \delta_{k k(\boldsymbol{z}^+)} \delta_{k k(\boldsymbol{z})}$, with the relaxed hard cluster assignment via a classification head, the objective for the **inter-view** clustering can be expressed by

$$\min_{\mathcal{M}} \frac{1}{N'} \sum_{\hat{\boldsymbol{z}}\in\hat{\mathcal{Z}}} \text{H}(\boldsymbol{p}(\hat{\boldsymbol{z}}^+), \boldsymbol{p}(\hat{\boldsymbol{z}})) + D_{\text{KL}}\left(\bar{\boldsymbol{p}} \| \boldsymbol{\pi}\right), \tag{11}$$

where $\boldsymbol{p}(\boldsymbol{x}) = \text{softmax}\left(\tau' \boldsymbol{W}_C^\top \boldsymbol{x}\right)$, $\tau' \ll 1$, $\boldsymbol{W}_C$ is a matrix of K orderly concatenated centroids, and $\text{H}(x, y) = -x \log y$ (cf. Appendix C).

Positive sample retrieval Unlike the common instance-level SSL, the positive samples in FLSL are amorphous clusters of features, $(\tilde{\boldsymbol{z}}^+, \tilde{\boldsymbol{z}})$, corresponding to the same local semantics in two views. In contrast to previous works assigning the best-matching patch [32, 51] or thresholded vicinity [58], we leverage the cluster assignment mechanism inherent in *mean-shift*, where a query $\boldsymbol{z}$ is automatically assigned to a cluster represented by the return $\hat{\boldsymbol{z}}$. For query from another view, the *mean-shift* naturally manifests as a cross-attention (CA),

$$\hat{\boldsymbol{z}}^+ = \boldsymbol{Z}^+ \operatorname{softmax}\left(\tau \boldsymbol{z}^\top \boldsymbol{Z}^+\right), \tag{12}$$

For locally and globally semantic representations, the returned representative $\hat{\boldsymbol{z}}^+$ of the cluster from the augmented view $\boldsymbol{Z}^+$ should agree with representative $\hat{\boldsymbol{z}}$ of the cluster containing the query $\boldsymbol{z}$. The process can be viewed as data retrieval in dense associative memory recognized in [41]

4.3 FLSL Objective

By combining the objectives from the two clustering levels, we arrive at the objective of FLSL:

$$\min \frac{1}{N'} \sum_{\boldsymbol{Z}\in\mathcal{Z}} \sum_{\boldsymbol{z}\in\boldsymbol{Z}} \upsilon \|\boldsymbol{z}-\hat{\boldsymbol{z}}\|_2^2 + \eta \sum_{\boldsymbol{z}\in\boldsymbol{Z}} \mathrm{H}(\boldsymbol{p}(\hat{\boldsymbol{z}}^+), \boldsymbol{p}(\hat{\boldsymbol{z}})) + \gamma D_{\mathrm{KL}}\left(\bar{\boldsymbol{p}} \| \boldsymbol{\pi}\right), \tag{13}$$
$$\text{with } \hat{\boldsymbol{z}} = \mathrm{SA}(\boldsymbol{z}, \boldsymbol{Z}, \boldsymbol{Z}),\ \hat{\boldsymbol{z}}^+ = \mathrm{CA}(\boldsymbol{z}, \boldsymbol{Z}^+, \boldsymbol{Z}^+),$$

where υ, η and γ are the hyperparameters controlling the importance of each term, and both SA and CA are non-parametric mean shift.

Figure 2 illustrates the FLSL framework. We follow the common joint-embedding strategy of SSL, except that we simultaneously maximize the agreement between the probability vectors of the positive cluster representatives $(\boldsymbol{p}(\hat{\boldsymbol{z}}^+), \boldsymbol{p}(\hat{\boldsymbol{z}}))$ and the agreement between the cluster members and their representative $(\boldsymbol{z}, \hat{\boldsymbol{z}})$. The KL-divergence term in Eq. 13 serves as a volume maximization regularizer. In our experiments, we use a uniform prior $\boldsymbol{\pi} = 1/K$. Experiments show that the FLSL objective effectively promote locally and globally semantic representations, resulting in significantly improved transferability of learnt features to object detection and segmentation. Note that FLSL does not involve a class token in its objective (Eq. 13) since it is a self-supervised learning method for dense prediction tasks.

5 Experiments

In this section, we evaluate the performance of FLSL by conducting extensive experiments. Specifically, we compare FLSL to existing SSL approaches on multiple dense prediction benchmarks: (i) MS-COCO [35] object detection and instance segmentation, (ii) UAVDT [20] object detection from UAV platforms, and (iii) DAVIS video instance segmentation [39]. Moreover, we investigate the properties of FLSL features in terms of semantic alignment and feature separability in the embedding space. Detailed experimental setups are provided in the respective subsections and supplementary materials. All our experiments are performed on Nvidia RTX A6000. Our source code can be found at `https://github.com/ISL-CV/FLSL.git`.

Implementation details The implementation of ViT in our experiments mostly follows DeiT [46] excluding the `[class]` token. The configuration of the ViT variants utilized in this paper is summarized in Appendix D. The coefficients of Eq. 13 in our experiments are $\upsilon = 0.3$, $\eta = 1$ and $\gamma = 5$ unless stated otherwise. We set the number of centroids $K = 4,096$, and assume a uniform prior, *i.e.*, $\pi_k = 1/K,\ \forall k$. Models are pretrained on ImageNet-1k [45] dataset using AdamW optimizer [38] with a batch size of 512. We follow the data augmentation from BYOL [23] (*e.g.*, color jittering of brightness, contrast, saturation and hue, Gaussian blur and solarization) with preceding random crops and resizing (to 224×224) and make them asymmetric. Contrasting among dense features can be computationally expensive. Therefore, we apply a random pooling in a 2×2 grid to the queries. All ViT models are pretrained for 300 epochs as in most baselines for a fair comparison. FLSL pseudo-code, complete training details, and settings of augmentation pipeline are provided in Appendix D.

Baselines We compare FLSL with various existing SSL approaches that are based on the ResNet [27] and ViT [19] architectures: (a) self-supervised ResNet: MoCo-v2 [12], DetCo [55], DenseCL [51], BYOL [23], and SCRL [43]; and (b) self-supervised ViT: MoCo-v3 [13], MoBY [57], DINO [8], MAE [24], SelfPatch [60], and ADCLR [61].

Pretrain	Backbone	Epoch	#Params	AP^{bbox}	AP^{bbox}_{50}	AP^{bbox}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{70}
MoCo-v2	RN50	200	23M	38.9	59.2	42.4	35.5	56.2	37.8
DetCo	RN50	200	23M	40.1	61.0	43.9	36.4	58.0	38.9
DenseCL	RN50	200	23M	40.3	59.9	44.3	36.4	57.0	39.2
BYOL	RN50	1000	23M	40.4	61.6	44.1	37.2	58.8	39.8
SCRL	RN50	1000	23M	41.3	62.4	45.0	37.7	59.6	40.7
MOCO-v3	ViT-S/16	300	21M	39.8	62.6	43.1	37.1	59.6	39.2
MoBY	ViT-S/16	300	21M	41.1	63.7	44.8	37.6	60.3	39.8
DINO	ViT-S/16	300	21M	40.8	63.4	44.2	37.3	59.9	39.5
DINO+SelfPatch	ViT-S/16	200	21M	42.1	64.9	46.1	38.5	61.3	40.8
ADCLR	ViT-S/16	300	21M	44.3	65.4	47.6	39.7	62.1	41.5
FLSL	ViT-S/16	300	21M	44.9	66.1	48.1	40.8	64.7	44.2
FLSL	ViT-S/8	300	21M	46.5	69.0	51.3	42.1	65.3	45.0

Table 1: MASK R-CNN ON COCO

Pretrain	AP^{bbox}	AP^{bbox}_{s}	AP^{bbox}_{m}	AP^{bbox}_{l}	AP^{mk}
None	48.1	-	-	-	42.6
IN-1k Supv.	47.6	-	-	-	42.4
IN-21k Supv.	47.8	-	-	-	42.6
IN-1k DINO	48.9	32.9	52.2	62.4	43.7
IN-1k MAE	51.2	34.9	54.7	66.0	45.5
IN-1k FLSL	53.1	36.9	56.2	67.4	47.0

Table 2: VITDET-B/16 WITH MASK R-CNN ON COCO

Pretrain	Backbone	AP_{VOC}
IN-1k DINO	ViT-S/16	48.9
IN-1k DINO	ViT-B/16	49.1
IN-1k DINO	ViT-S/8	51.1
IN-1k FLSL	ViT-S/16	53.1
IN-1k FLSL	ViT-B/16	53.5
IN-1k FLSL	ViT-S/8	55.2

Table 3: FASTER R-CNN FPN ON UAVDT

Protocol for hyperparameter tuning Standard instance-level SSL evaluation protocols typically utilize one of the two approaches: employing a k-NN classifier or training a linear classifier on fixed features. Since FLSL learns dense semantic representations rather than a single instance-level representation, both standard evaluation protocols are not suitable for evaluating FLSL in training. Moreover, fine-tuning on a downstream dense prediction tasks can be computationally expensive due to complex prediction heads, and may introduce task-specific biases during hyperparameter tuning. Therefore, we design a bbox-aligned k-NN classifier modified from [54] to evaluate the feature quality directly without additional network tuning. Here is an overview of the method. Features of the training data are first extracted with a fixed model. These features are then aligned with their corresponding bounding boxes provided by ILSVRC [44]. For each image, a certain number of representative features (*e.g.*, 9) are selected by a partition criterion and stored in memory. The k-NN classifier matches each selected features to its k-nearest stored features, which collectively vote for its label. A feature is considered successfully classified if any of the representative features match its class. This protocol is employed for hyperparameter tuning and ablation study of the FLSL pipeline. Appendix E provides further details on the choice of k, implementation specifics and evaluation results.

5.1 MS-COCO Object Detection & Segmentation

We adopt Mask R-CNN detection framework by incorporating three variants of ViT: (i) ViT-S/16 with FPN [34], (ii) ViT-S/8 with FPN, and (iii) ViT-B/16 with simple feature pyramid (ViTDet) [33]. Models of (i) and (ii) are fine-tuned following the multi-scale training [53, 6] under the standard $1\times$ schedule for a fair comparison. For the model of (iii), we follow the training recipe of [33] and fine-tune the model for 100 epochs.

Results. Table 1 reports the detection and segmentation performance of ViT-S/16 and ViT-S/8 with Mask R-CNN [26] on COCO. Specifically, FLSL with ViT-S/16 outperforms ADCLR [61] by +0.6% and +1.1%, and substantially outperforms DINO+SelfPatch [60] by +2.8% and +2.4% on detection (AP^{bbox}) and segmentation (AP^{mk}), respectively. Both baseline methods feature patch-level contrastive learning. Unlike SelfPatch contrasting between patches within the adjacent neighborhood and ADCLR contrasting via learned queries of random crops, FLSL contrasts the representatives (modes) of semantic cluster of features, which aligns closer with the downstream tasks and thus leads to superior performance. Notably, FLSL with ViT-S/8 further improves the performance by a large margin of +4.4% in AP^{bbox} and +3.6% AP^{mk} over SelfPatch. Table 2 summarizes the results of ViTDet. FLSL shows large performance gains over the DINO baseline by +4.2% AP^{bbox} and +3.3% AP^{mk}. FLSL also outperforms the SOTA generative approach, MAE, by +1.7% and +1.4% in the two tasks, respectively.

5.2 Small Object Detection: UAVDT

To assess the transferability of FLSL beyond the datasets of common images like COCO, we further investigate its performance on a UAV benchmark, UAVDT [20], which exhibits significant domain shifts from common images (*i.e.*, images captured by ground-level cameras). We utilize Faster

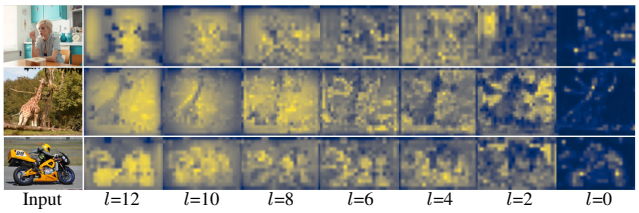


Figure 3: Visualization of the maps of the *aggregated attention score* (AAS) from different layers of ViT-S/16. $l = 0$ denotes the projection layer. As layer goes deeper, the map becomes more partitioned with brightness aligned with the area of the underlying semantic region, *e.g.*, objects or stuff.

Pretrain	Arch.	$(\mathcal{J}\&\mathcal{F})_m$	$\mathcal{J}_m$	$\mathcal{F}_m$
IN-1k supv.	ViT-S/8	66.0	63.9	68.1
VLOG CT	RN50	48.7	46.4	50.0
YT-VOS MAST	RN18	65.5	63.3	67.6
IN-1k DINO	ViT-S/16	61.8	60.2	63.4
IN-1k DINO	ViT-B/16	62.3	60.7	63.9
IN-1k DINO	ViT-S/8	69.9	66.6	73.1
IN-1k FLSL	ViT-S/16	65.6	62.4	69.4
IN-1k FLSL	ViT-B/16	66.1	62.9	70.0
IN-1k FLSL	ViT-S/8	73.5	69.9	78.1

Table 4: DAVIS 2017 VIDEO INSTANCE SEGMENTATION. We evaluate the quality of frozen features on video instance tracking. We report mean region similarity $\mathcal{J}_m$ and mean contour-based accuracy $\mathcal{F}_m$.

R-CNN framework [42] with the same ViT variants used in the COCO experiments and follow the training settings outlined in ClusDet [59]. All ViT-backboned models are trained with $1\times$ schedule.

Result Table 3 presents the performance of ViT-S/16, ViT-S/8, and ViT-B/16 with Faster R-CNN for detection tasks on UAVDT under different pretrain schemes. We utilize the official evaluation method in [20], which calculates the class-agnostic VOC AP exclusive of the predictions that falls in the ignored areas. FLSL consistently outperforms DINO (a typical instance-level SSL for ViT) across all three ViT variants by a significant margin. With smaller objects and an imbalanced foreground-background ratio, the significance of local semantics becomes evident. Models require local context to discover small objects and make accurate predictions rather than relying solely on the global semantics of the entire image. This situation aligns well with the strengths of FLSL.

5.3 DAVIS Segmentation

To further assess the quality of frozen features learned by FLSL, we evaluate FLSL-pretrained ViT models on DAVIS2017 [39], following the evaluation protocol in [30, 8] that requires fixed representations with no extra training.

Results Table 4 shows that FLSL consistently outperforms DINO across all ViT variants in our experiments. The protocol evaluates the quality of learned dense features via segmenting scenes with k-nearest neighbors ($k = 5$) within a fixed window (12×12) between consecutive frames. This requires dense features to be locally semantic, *i.e.*, features corresponding to the same semantics should be more correlated. Therefore, the improved performance confirms that FLSL encourages model to extract locally semantic representations.

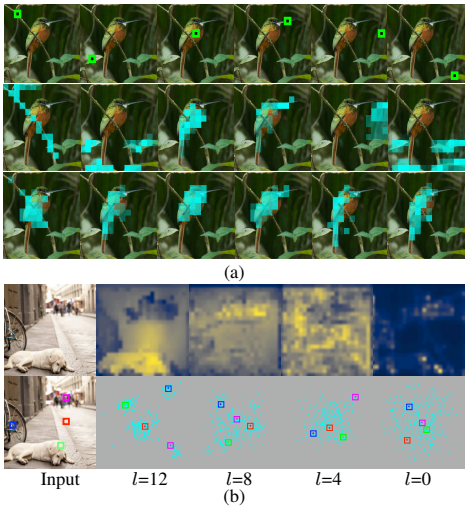


Figure 4: (a) visualization of top-10% patches obtained by thresholding the self-attention maps of query patches (top) in the last layer of ViT-S/16 trained with FLSL (middle) and with DINO (bottom). FLSL encourages the model to learn semantic correlations among patches; (b) visualization of separability of the patch representations of an image throughout the transformer (ViT-S/16).

5.4 Alignment with Image Semantics

To show that FLSL is better aligned with the semantic layout of an image than the common SSL methods, Figure 4 compares the self-attention maps from the last layer of a ViT-S/16 trained with FLSL to those of DINO. The query tokens are the patches in the last ViT layer. The visualizations are obtained with 224^2 images. The attention segmentation is obtained by thresholding the self-attention map to keep the top-10% of the mass. As shown in the middle and bottom rows of Figure 4(a), DINO promotes object-centered attention (*i.e.*, class related content is dominating), while FLSL encourages attention to the regions of high semantic correlation with the query and results in masks consistent with the objects/stuff.

5.5 Feature Distribution and Separability

We demonstrate the qualitative results by visualizing the aggregated attention score (AAS) and the feature distribution in the embedding space through t-sne [48] in Figure 3 and Figure 4(b), respectively. To generate the map of AAS, we sum up all the self-attention maps, normalize the resulting map

Sinkhorn	η	γ	$\upsilon = 0.0$	$\upsilon = 0.1$	$\upsilon = 0.2$	$\upsilon = 0.3$	$\sim$	$\upsilon = 1.0$
✓	1.0	1.0	0.1	68.7	70.7	71.2	$\sim$	65.1
×	1.0	1.0	-	-	-	66.6	-	-
×	1.0	5.0	-	-	-	72.4	-	-

Table 5: IMPACT OF COEFFICIENTS IN THE FLSL OBJECTIVE.

K	1024	2048	4096	8192	16384
k-NN top-1	68.1	72.1	72.4	72.5	72.1

Table 6: IMPACT OF NUMBER OF CENTROIDS K

with its maximum score and visualize it as a thermal image, *i.e.*, the brighter the pixel, the higher the score. For a semantically well-separated image, each patch only attends to the patches of its own semantic region, *e.g.*, a patch of an object has high attention scores only with the patches of that object and low scores with the rest. This results in an image with partitions of different brightness proportional to the area of that region, *i.e.*, the larger the size of an object/stuff, the brighter the color. As shown in Figure 3, as the layer goes deeper, the brightness partition of the AAS is more consistent with the objects and stuff in the images (*e.g.*, person, giraffes, motorcycles, greens, wall, and ground, etc.), which indicates the desired separation of the learned features. This is also reflected in the t-sne visualization of the embeddings in Figure 4(b), where the representations become more clustered and separated as the attention layer goes deeper.

5.6 Ablation Study

Due to limited space, we present two major ablation studies in this section to help understand the effectiveness of FLSL. The model considered for this entire study is ViT-S trained with 100 epochs. We refer the reader to Appendix G for the complete work.

Impact of coefficients in the FLSL objective The FLSL objective (Eq. 13) contains three components: (1) similarity between ℓ_2-normalized $\boldsymbol{z}$ (features) and $\hat{\boldsymbol{z}}$ (modes), (2) cross-entropy of the probabilities of an augmented pair $H(p(\hat{\boldsymbol{z}}^+), p(\hat{\boldsymbol{z}}))$, and (3) the volume maximization regularizor $D_{\text{KL}}\left(\bar{\boldsymbol{p}} \| \boldsymbol{\pi}\right)$. It is computationally expensive to optimally determine the values of more than two coefficients by performing grid search, especially when the ratios among them are large. We tackle this problem by first fixing $\eta = 1$ and setting $\gamma = 1$ along with Sinkhorn normalization [16] to perform a grid search on the value of υ with the empirical base condition $\upsilon \leq 1$ and $\gamma \geq 1$ [60, 1]. With the fixed υ, we then perform another grid search on γ without Sinkhorn normalization. We implement Sinkhorn normalization as the softmax operation along the batch dimension. Table 5 summerizes the score of bbox-aligned k-NN evaluation using different coefficient settings.

Impact of number of centroids K FLSL is formulated as an explicit clustering problem, with the output dimension of the last fully-connected layer equal to the number of centroids K. Compared to its instance-level counterpart DINO [8], FLSL enjoys a smaller output dimension (shown in Table 6). This is because images have higher feature variance compared to feature clusters. For example, an image in ImageNet may contain diverse content from different categories, requiring a large number of centroids to cover the distribution. In contrast, a semantic cluster contains highly correlated features, such as similar textures or objects from the same category, thus requiring fewer centroids. Experimentally, we find that a large number of centroids benefits performance, but is detrimental and costly when being too large. We pick $K = 4,096$ for all our experiments as it strikes a good balance between performance and cost-effectiveness.

Other ablations including the impact of batch size and random pooling window size are relegated to Appendix.

6 Conclusions

This paper proposes FLSL, a feature-level self-supervised learning method that bridges the gap between the current SSL methods and downstream dense prediction tasks. We demonstrate for the first time the underlying *mean-shift* clustering process of ViT, which aligns well with natural image semantics. Facilitated by ViT for joint embedding and feature clustering, FLSL performs a two-level clustering: (i) intra-view clustering to extract the representatives for clusters of features within an image, and (ii) inter-view clustering to encourage the representatives to be globally semantic over the entire dataset. FLSL achieves a significant improvement over the SOTAs in the dense prediction tasks, including object detection and instance segmentation.

Limitations and broader impacts FLSL does not have any significant limitations other than the method is more complex (due to its two-level clustering) than other SSL methods, and it currently only fits for ViT-based models on dense prediction tasks. Exploring ways to extend FLSL for tasks that necessitate a global representation while retaining its existing properties could be a potential future work. As far as we can foresee, there is no negative societal impact.

7 Acknowledgment

This research was sponsored by the Army Research Laboratory under Cooperative Agreement #W911NF-22-2-0025. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Mike Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. In *European Conference on Computer Vision*, pages 456–473. Springer, 2022.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

[4] Paul S Bradley, Kristin P Bennett, and Ayhan Demiriz. Constrained k-means clustering. *Microsoft Research, Redmond*, 20(0):0, 2000.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

[8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[9] Miguel Carreira-Perpiñán. Reconstruction of sequential data with probabilistic models and continuity constraints. *Advances in neural information processing systems*, 12, 1999.

[10] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020.

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[12] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[13] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021.

[14] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE transactions on pattern analysis and machine intelligence*, 17(8):790–799, 1995.

[15] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.

[16] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[18] Jian Ding, Enze Xie, Hang Xu, Chenhan Jiang, Zhenguo Li, Ping Luo, and Gui-Song Xia. Deeply unsupervised patch re-identification for pre-training object detectors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[20] Dawei Du, Yuankai Qi, Hongyang Yu, Yifan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian. The unmanned aerial vehicle benchmark: Object detection and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 370–386, 2018.

[21] Benjamin L Edelman, Surbhi Goel, Sham Kakade, and Cyril Zhang. Inductive biases and variable creation in self-attention mechanisms. In *International Conference on Machine Learning*, pages 5793–5831. PMLR, 2022.

[22] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[23] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[24] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[25] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[26] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[28] Christian Hennig, Marina Meila, Fionn Murtagh, and Roberto Rocci. *Handbook of cluster analysis*. CRC Press, 2015.

[29] Ashraful Islam, Benjamin Lundell, Harpreet Sawhney, Sudipta N Sinha, Peter Morales, and Richard J Radke. Self-supervised learning with local contrastive loss for detection and semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5624–5633, 2023.

[30] Allan Jabri, Andrew Owens, and Alexei Efros. Space-time correspondence as a contrastive random walk. *Advances in neural information processing systems*, 33:19545–19560, 2020.

[31] Dmitry Krotov and John Hopfield. Large associative memory problem in neurobiology and machine learning. *arXiv preprint arXiv:2008.06996*, 2020.

[32] Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient self-supervised vision transformers for representation learning. *arXiv preprint arXiv:2106.09785*, 2021.

[33] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*, 2022.

[34] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[35] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[36] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[37] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[38] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[39] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[40] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *OpenAI*, 2018.

[41] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020.

[42] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[43] Byungseok Roh, Wuhyun Shin, Ildoo Kim, and Sungwoong Kim. Spatially consistent representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1144–1153, 2021.

[44] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[45] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[46] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[47] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[48] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[50] Matt P Wand and M Chris Jones. *Kernel smoothing*. CRC press, 1994.

[51] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3024–3033, 2021.

[52] Fangyun Wei, Yue Gao, Zhirong Wu, Han Hu, and Stephen Lin. Aligning pretraining for detection via object-level contrastive learning. *Advances in Neural Information Processing Systems*, 34:22682–22694, 2021.

[53] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[54] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018.

[55] Enze Xie, Jian Ding, Wenhai Wang, Xiaohang Zhan, Hang Xu, Peize Sun, Zhenguo Li, and Ping Luo. Detco: Unsupervised contrastive learning for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8392–8401, 2021.

[56] Jiahao Xie, Xiaohang Zhan, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Unsupervised object-level representation learning from scene images. *Advances in Neural Information Processing Systems*, 34:28864–28876, 2021.

[57] Zhenda Xie, Yutong Lin, Zhuliang Yao, Zheng Zhang, Qi Dai, Yue Cao, and Han Hu. Self-supervised learning with swin transformers. *arXiv preprint arXiv:2105.04553*, 2021.

[58] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16684–16693, 2021.

[59] Fan Yang, Heng Fan, Peng Chu, Erik Blasch, and Haibin Ling. Clustered object detection in aerial images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8311–8320, 2019.

[60] Sukmin Yun, Hankook Lee, Jaehyung Kim, and Jinwoo Shin. Patch-level representation learning for self-supervised vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8354–8363, 2022.

[61] Shaofeng Zhang, Feng Zhu, Rui Zhao, and Junchi Yan. Patch-level contrasting without patch cor-respondence for accurate and dense con-trastive representation learning. 2023.

[62] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.

FLSL: Feature-level Self-supervised Learning

Supplementary Materials

A Intra-veiw clustering with mean-shift

An image can be represented as an empirical probability density function that comprises amorphous clusters of features. Given a dense representation of an image $\boldsymbol{Z} = \{\boldsymbol{z}_i\}_{i=1}^N$ and the *mean-shift* clustering scheme, the posterior of $\boldsymbol{z}_j$ given $\boldsymbol{z}_i$ indicates the probability of feature $\boldsymbol{z}_i$ being assigned to the cluster of $\boldsymbol{z}_j$, which is defined as follows:

$$
\begin{aligned}
p(\boldsymbol{z}_j|\boldsymbol{z}_i) &= \left[\text{softmax}\left(\tau \boldsymbol{z}_i^\top \boldsymbol{Z}\right)\right]_j \qquad (14)\\
&= e^{\tau \boldsymbol{z}_i^\top \boldsymbol{z}_j} \Bigg/ \left(\sum_{k\in c_i} e^{\tau \boldsymbol{z}_i^\top \boldsymbol{z}_k} + \sum_{k\in[N]\setminus c_i} e^{\tau \boldsymbol{z}_i^\top \boldsymbol{z}_k} \right)\\
&= 1 \Bigg/ \left(\left(\sum_{k\in c_i} e^{-(\boldsymbol{z}_i^\top \boldsymbol{z}_j - \boldsymbol{z}_i^\top \boldsymbol{z}_k)\tau} \right) + \sum_{k\in[N]\setminus c_i} e^{-(\boldsymbol{z}_i^\top \boldsymbol{z}_j - \boldsymbol{z}_i^\top \boldsymbol{z}_k)\tau} \right)\\
&\geq 1 \Bigg/ \left(\left(\sum_{k\in c_i} e^{-(\boldsymbol{z}_i^\top \boldsymbol{z}_j - \boldsymbol{z}_i^\top \boldsymbol{z}_k)\tau} \right) + (N - |c_i|) e^{-\Delta_{ij}\tau} \right), \qquad (15)
\end{aligned}
$$

where τ is the inverse temperature, c_i is the set of indices of points contained in the cluster of $\boldsymbol{z}_i$, $[N] = \{1, \ldots, N\}$, and Δ_{ij} is the cluster separation with respect to $\boldsymbol{z}_i$, defined as

$$
\Delta_{ij} = \boldsymbol{z}_i^\top \boldsymbol{z}_j - \max_{m\in[N]\setminus c_i} \boldsymbol{z}_i^\top \boldsymbol{z}_m, \; j \in c_i, \qquad (16)
$$

measuring the gain of similarity between $\boldsymbol{z}_i$ and an in-cluster point $\boldsymbol{z}_j$ over the similarity between $\boldsymbol{z}_i$ and the out-cluster point $\boldsymbol{z}_k$ that is closest to $\boldsymbol{z}_i$.

To achieve locally semantic representations, our objective is for the points within each cluster to be in close proximity to each other or, equivalently, close to their cluster representative. This proximity ensures consistency in encoded semantics. Additionally, we aim for these in-cluster points to be distinctly separated from the points outside the cluster. This separation encourages well-defined clusters to accurately reflect different semantics, i.e., a large Δ_{ij} and a small in-cluster variance. As Δ becomes sufficiently large (with a proper inverse temperature), the RHS of Eq. 15 can be approximated as $1/\sum_{k\in c_i} e^{-(\boldsymbol{z}_i^\top \boldsymbol{z}_j - \boldsymbol{z}_i^\top \boldsymbol{z}_k)\tau}$ for in-cluster points, $i, j \in c_i$. Meanwhile, the posterior for the out-cluster points, $p(\boldsymbol{z}_{j\notin c_i}|\boldsymbol{z}_i)$, approaches 0 at the rate of

$$
p(\boldsymbol{z}_{j\notin c_i}|\boldsymbol{z}_i) \;\leq\; 1 \Bigg/ \left(\left(\sum_{k\in c_i} e^{\tau \min_{k\in c_i} \Delta_{ik}} \right) + (N - |c_i|) e^{-\tau \max_{k\notin c_i} (\boldsymbol{z}_i^\top \boldsymbol{z}_j - \boldsymbol{z}_i^\top \boldsymbol{z}_k)} \right). \qquad (17)
$$

The resulting return of a single *mean-shift* update becomes

$$
\hat{\boldsymbol{z}}_i = \boldsymbol{Z}\text{softmax}\left(\tau \boldsymbol{z}_i^\top \boldsymbol{Z}\right) = \sum_{j\in[N]} p(\boldsymbol{z}_j|\boldsymbol{z}_i)\boldsymbol{z}_j \approx \sum_{j\in c_i} \frac{1}{\sum_{k\in c_i} e^{-(\boldsymbol{z}_i^\top \boldsymbol{z}_j - \boldsymbol{z}_i^\top \boldsymbol{z}_k)\tau}} \boldsymbol{z}_j + \boldsymbol{0}, \qquad (18)
$$

which is essentially a weighted sum of the in-cluster points **only**. To promote the aforementioned property while maintaining low in-cluster variance, one approach is to drive the point closer to its cluster representative by optimizing

$$
\min \sum_{i=1}^{N} \|\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i\|_2^2, \qquad \text{with } \hat{\boldsymbol{z}}_i = \boldsymbol{Z}\text{softmax}(\tau \boldsymbol{z}_i^\top \boldsymbol{Z}). \qquad (19)
$$

Notably, with a large inverse temperature $\tau \gg 1$, a single *mean-shift* update becomes the single-step pattern retrieval mechanism in dense associative memory (DAM) [31, 41].

B The GMM formulation of the constrained k-means objective

The *k-means* objective with generalized non-empty cluster constraint [4] can be expressed as

$$\min_{\mathcal{M}} \frac{1}{N'} \sum_{\hat{\boldsymbol{z}} \in \hat{\mathcal{Z}}} \sum_{k=1}^{K} \delta_{k k(\hat{\boldsymbol{z}})} \|\hat{\boldsymbol{z}} - \boldsymbol{\mu}_{k(\hat{\boldsymbol{z}})}\|_2^2 + D_{\mathrm{KL}}\left(\bar{\boldsymbol{p}} \| \boldsymbol{\pi}\right), \tag{20}$$

where $\mathcal{M}$ is a set of K centroids $\{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K\}$, $\hat{\mathcal{Z}}$ is a set of cluster representatives over the entire dataset, $N' = |\hat{\mathcal{Z}}|$, $k(\hat{\boldsymbol{z}}) = \arg\min_k \|\boldsymbol{\mu}_k - \hat{\boldsymbol{z}}\|_2$, δ_{ij} is the *Kronecker delta*, with $\delta_{ij} = 1$ iff $i = j$, and 0 otherwise, $[\bar{\boldsymbol{p}}]_{[i]} = {}^1\!/_{N'} \sum_{\hat{\boldsymbol{z}}} \delta_{i k(\hat{\boldsymbol{z}})}$, and $\boldsymbol{\pi}$ is the prior, *e.g.*, a vector of the preset proportion for each cluster.

As mentioned in the main paper, a common approach to tackle the optimization problem above is to relax the hard cluster assignment constraint $\delta_{ij} \in \{0, 1\}$ to $[0, 1]$ with a classification head to $\hat{\boldsymbol{z}}$. This relaxes Eq. 20 to the more general Gaussian Mixture Model (GMM) formulation, allowing each point to have a partial membership of each cluster with a certain probability. The GMM ELBO can be expressed by the average term-by-term reconstruction and KL to prior as

$$\mathcal{L}(\theta, \mathcal{M}, \boldsymbol{\Sigma}) = -\frac{1}{N'} \left(\sum_{\hat{\boldsymbol{z}} \in \hat{\mathcal{Z}}} \sum_{\boldsymbol{\mu} \in \mathcal{M}} q(\boldsymbol{\mu}|\hat{\boldsymbol{z}}) d(\hat{\boldsymbol{z}}, \boldsymbol{\mu}; \boldsymbol{\Sigma}_{\boldsymbol{\mu}}) + \sum_{\hat{\boldsymbol{z}} \in \hat{\mathcal{Z}}} D_{\mathrm{KL}}\left(q(\boldsymbol{\mu}|\hat{\boldsymbol{z}}) \| \boldsymbol{\pi}\right) \right) + C, \tag{21}$$

where $d(\boldsymbol{z}, \boldsymbol{\mu}; \boldsymbol{\Sigma}_{\boldsymbol{\mu}}) = (\boldsymbol{z} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}_{\boldsymbol{\mu}}^{-1} (\boldsymbol{z} - \boldsymbol{\mu})$ is the *Mahalanobis* distance, C is a constant under the assumption of homoscedastic and isotropic Gaussian kernel. With a classification head, the posterior of $\hat{\boldsymbol{z}}$ belonging to cluster k is

$$q(\boldsymbol{\mu}_k|\hat{\boldsymbol{z}}) = \left[\mathrm{softmax}\left(\tau' \boldsymbol{W}_{\mathcal{M}}^\top \hat{\boldsymbol{z}} + \log \boldsymbol{\pi} - \tau' \left(\hat{\boldsymbol{z}}^\top \hat{\boldsymbol{z}} + \mathrm{diag}\left(\boldsymbol{W}_{\mathcal{M}}^\top \boldsymbol{W}_{\mathcal{M}}\right)\right)\right)\right]_k, \tag{22}$$

where τ' is the inverse temperature, and $\boldsymbol{W}_{\mathcal{M}}$ is a matrix of K concatenated centroids with its kth column corresponding to $\boldsymbol{\mu}_k$. Particularly, we assume all vectors are ℓ_2-normalized. This further simplifies the posterior to $\boldsymbol{q}(\boldsymbol{\mu}|\hat{\boldsymbol{z}}) = \mathrm{softmax}(\tau' \boldsymbol{W}_{\mathcal{M}}^\top \hat{\boldsymbol{z}} + \log \boldsymbol{\pi})$, which conforms with the output of a classification head as a mixing proportion.

The hard cluster assignment in Eq. 20 can be recovered by sharpening the posterior with a small covariance, or equivalently, a large inverse temperature τ', *i.e.*,

$$\begin{aligned} \lim_{\tau' \to \infty} q_\phi(\boldsymbol{\mu}_k|\hat{\boldsymbol{z}}) &= \lim_{\tau' \to \infty} \left[\mathrm{softmax}(\tau' \boldsymbol{W}_{\mathcal{M}}^\top \hat{\boldsymbol{z}} + \log \boldsymbol{\pi})\right]_k \\ &= \lim_{\tau' \to \infty} \left[\mathrm{softmax}(\tau' \boldsymbol{W}_{\mathcal{M}}^\top \hat{\boldsymbol{z}})\right]_k = \delta_{k k(\hat{\boldsymbol{z}})}. \end{aligned} \tag{23}$$

With a sufficiently large inverse temperature, the KL-divergence term of Eq. 21 becomes

$$\frac{1}{N'} \sum_{\hat{\boldsymbol{z}} \in \hat{\mathcal{Z}}} D_{\mathrm{KL}}\left(\delta_{k k(\hat{\boldsymbol{z}})} \| \boldsymbol{\pi}\right) = -\sum_{k=1}^{K} \frac{N'_k}{N'} \log \boldsymbol{\pi}_k, \tag{24}$$

where $N'_k = \sum_{\hat{\boldsymbol{z}} \in \hat{\mathcal{Z}}} \mathbf{1}_{[\boldsymbol{k}(\hat{\boldsymbol{z}}) = k]}$. By defining $[\bar{\boldsymbol{p}}]_k = \frac{N'_k}{N'}$ and adding back the non-empty constraint as the negative entropy of $\bar{\boldsymbol{p}}$, the resulting GMM ELBO recovers Eq. 9 with $d(\hat{\boldsymbol{z}}, \boldsymbol{\mu}; \boldsymbol{\Sigma}_{\boldsymbol{\mu}}) \propto \|\hat{\boldsymbol{z}} - \boldsymbol{\mu}_{k(\hat{\boldsymbol{z}})}\|_2^2$.

C The cross-entropy formulation of the constrained k-means with positive samples

With positive pairs $(\hat{\boldsymbol{z}}^+, \hat{\boldsymbol{z}})$ created via data augmentation, the constrained *k-means* objective in Eq. 20 can be formulated as *k-means* clustering with an extra separation margin for $\hat{\boldsymbol{z}}^+$.

Here, we present the derivation of Eq. 11 in the main paper, considering a more general setting that involves multiple positive samples $\{\hat{\boldsymbol{z}}^{(a)}\}_{a=1}^{A}$ anchored on $\hat{\boldsymbol{z}}^{(0)} = \hat{\boldsymbol{z}}$ through data augmentation. The objective in Eq. 10 from the main paper is essentially a special case of the following expression, where the number of positive pairs A equal to 1:

$$\min_{\mathcal{M}} \frac{1}{N'} \sum_{\hat{\boldsymbol{z}} \in \hat{\mathcal{Z}}} \left(\sum_{k=1}^{K} \delta_{k k(\hat{\boldsymbol{z}})} \|\hat{\boldsymbol{z}} - \boldsymbol{\mu}_{k(\hat{\boldsymbol{z}})}\|_2^2 + \frac{1}{A} \sum_{a=1}^{A} \left(1 - \delta_{k(\hat{\boldsymbol{z}}^{(a)}) k(\hat{\boldsymbol{z}})}\right) \|\hat{\boldsymbol{z}}^{(a)} - \boldsymbol{\mu}_{k(\hat{\boldsymbol{z}})}\|_2^2 \right) + D_{\mathrm{KL}}\left(\bar{\boldsymbol{p}} \| \boldsymbol{\pi}\right), \tag{25}$$

which imposes that a point and its positive samples reside in the same cluster.

The above optimization problem can be tackled by minimizing its upper bound with a relaxed hard assignment. Specifically, the term inside the parenthesis is bounded by

$$\sum_{k=1}^{K} \delta_{kk(\hat{\boldsymbol{z}}^{(0)})} \|\hat{\boldsymbol{z}}^{(0)} - \boldsymbol{\mu}_{k(\hat{\boldsymbol{z}}^{0})}\|_2^2 + \frac{1}{A} \sum_{a=1}^{A} \left(1 - \delta_{k(\hat{\boldsymbol{z}}^{(a)})k(\hat{\boldsymbol{z}}^{(0)})}\right) \|\hat{\boldsymbol{z}}^{(a)} - \boldsymbol{\mu}_{k(\hat{\boldsymbol{z}}^{(0)})}\|_2^2$$

$$\leq \|\hat{\boldsymbol{z}}^{(0)} - \boldsymbol{\mu}_{k(\hat{\boldsymbol{z}}^{(0)})}\|_2^2 + \frac{1}{A} \max_{a \in A} \|\hat{\boldsymbol{z}}^{(a)} - \boldsymbol{\mu}_{k(\hat{\boldsymbol{z}}^{(0)})}\|_2^2 \sum_{a=1}^{A} \left(1 - \delta_{k(\hat{\boldsymbol{z}}^{(a)})k(\hat{\boldsymbol{z}}^{(0)})}\right). \tag{26}$$

By rewriting $1 - \delta_{k(\hat{\boldsymbol{z}}^{(a)})k(\hat{\boldsymbol{z}}^{(0)})}$ as $\sum_{k=1}^{K} \left(\delta_{kk(\boldsymbol{z}^{(a)})} - \delta_{kk(\hat{\boldsymbol{z}}^{(a)})}\delta_{kk(\hat{\boldsymbol{z}}^{(0)})}\right)$, the RHS of Eq. 26 becomes

$$\|\hat{\boldsymbol{z}}^{(0)} - \boldsymbol{\mu}_{k(\hat{\boldsymbol{z}}^{(0)})}\|_2^2 + \frac{1}{A} \max_{a \in A} \|\hat{\boldsymbol{z}}^{(a)} - \boldsymbol{\mu}_{k(\hat{\boldsymbol{z}}^{(0)})}\|_2^2 \sum_{a=1}^{A} \left(\sum_{k=1}^{K} \delta_{kk(\hat{\boldsymbol{z}}^{(a)})} - \sum_{k=1}^{K} \delta_{kk(\hat{\boldsymbol{z}}^{(a)})}\delta_{kk(\hat{\boldsymbol{z}}^{(0)})} \right)$$

$$= \|\hat{\boldsymbol{z}}^{(0)} - \boldsymbol{\mu}_{k(\hat{\boldsymbol{z}}^{(0)})}\|_2^2 + \frac{1}{A} \max_{a \in A} \|\hat{\boldsymbol{z}}^{(a)} - \boldsymbol{\mu}_{k(\hat{\boldsymbol{z}}^{(0)})}\|_2^2 \sum_{a=1}^{A} \sum_{k=1}^{K} \left(\delta_{kk(\hat{\boldsymbol{z}}^{(a)})} \left(1 - \delta_{kk(\hat{\boldsymbol{z}}^{(0)})}\right)\right), \tag{27}$$

which is bounded by

$$\leq \|\hat{\boldsymbol{z}}^{(0)} - \boldsymbol{\mu}_{k(\hat{\boldsymbol{z}}^{(0)})}\|_2^2 + \frac{1}{A} \max_{a \in A} \|\hat{\boldsymbol{z}}^{(a)} - \boldsymbol{\mu}_{k(\hat{\boldsymbol{z}}^{(0)})}\|_2^2 \sum_{a=1}^{A} \sum_{k=1}^{K} -\delta_{kk(\hat{\boldsymbol{z}}^{(a)})} \log\left(\delta_{kk(\hat{\boldsymbol{z}}^{(0)})} + \epsilon\right), \tag{28}$$

with $0 < \epsilon \ll 1$.

To our interest, we assume all vectors are ℓ_2-normalized. Thus, the bound in Eq. 28 can be further simplified to

$$4 + 4\frac{1}{A} \sum_{a=1}^{A} \sum_{k=1}^{K} -\delta_{kk(\hat{\boldsymbol{z}}^{(a)})} \log\left(\delta_{kk(\hat{\boldsymbol{z}}^{(0)})} + \epsilon\right). \tag{29}$$

By relaxing the hard assignment $\delta_{kk(\hat{\boldsymbol{z}})} \in \{0, 1\}$ to $[0, 1]$ using a classification head to $\hat{\boldsymbol{z}}$ as in the GMM formulation in Appendix B with a sufficiently large inverse temperature $\tau' \gg 1$, the optimization in Eq. 25 can be approached by

$$\min_{\mathcal{M}} \frac{1}{AN'} \sum_{a=1}^{A} \sum_{\hat{\boldsymbol{z}} \in \hat{\mathcal{Z}}} \mathrm{H}(\boldsymbol{p}(\hat{\boldsymbol{z}}^{(a)}), \boldsymbol{p}(\hat{\boldsymbol{z}})) + D_{\mathrm{KL}}\left(\bar{\boldsymbol{p}} \| \boldsymbol{\pi}\right), \tag{30}$$

where $\boldsymbol{p}(\hat{\boldsymbol{z}}) = \boldsymbol{q}(\boldsymbol{\mu}|\hat{\boldsymbol{z}}) = \mathrm{softmax}\left(\tau' \boldsymbol{W}_{\mathcal{M}}^{\top} \hat{\boldsymbol{z}}\right)$, and $\mathrm{H}(\boldsymbol{x}, \boldsymbol{y}) = -\boldsymbol{x}^{\top} \log \boldsymbol{y}$. When $A = 1$, *i.e.*, only considering a single positive pair, the above objective degenerates to Eq. 11 in the main paper.

model	#blocks	dim	#heads	#tokens	#params	im/s
ViT-S/16	12	384	6	196	21M	1,007
ViT-S/8	12	384	6	785	21M	180
ViT-B/16	12	768	12	196	85M	312

Table 8: VIT CONFIGURATION

D Implementation details

D.1 Network configuration

We follow the implementation used in DeiT [46] for all the ViT variants used in our experiments, and their configurations are summarized in Table 8.

In the table, "#blocks" is the number of transformer blocks, "dim" is the channel dimension, "#heads" is the number of heads in multi-head attention, "#tokens" is the length of the token sequence when considering 224^2 resolution inputs, "#params" is the total number of parameters (without counting the projection head), and "im/s" is the inference speed on a NVIDIA V100 GPU with 128 samples per forward.

D.2 Training details

The implementation of ViT in our experiments mostly follows DeiT [46], with the exception of excluding the `[class]` token. During pretext training, we set the coefficients in the FLSL objective as follows: $\upsilon = 0.3$, $\eta = 1.0$, and $\gamma = 5.0$, and assume a uniform prior, *i.e.*, $\pi_k = 1/K,\ \forall k$, with the number of centroids $K = 4096$. We pretrain the models on ImageNet-1k dataset without labels using AdamW optimizer [38] and a batch size of 512. In line with DINO, the learning rate linearly ramps up during the first 10 epochs to the base value determined with the linear scaling rule [22]: *lr*$= 0.0005$ with the reference *batch_size*$= 256$. The warm-up is followed by the learning rate decay governed by cosine schedule [37] with the target learning rate 10^{-6}. The weight decay also governed by a cosine schedule from 0.05 to 0.5. The update rule for teacher network is $\theta_t \leftarrow \lambda\theta_t + (1-\lambda)\theta_s$, with λ following a cosine schedule from 0.996 to 1. The inverse temperature for student classification head, τ_s, is set to $1/0.1$, while the inverse temperature for teacher classification head, τ_t, follows a linear warm-up from $1/0.04$ to $1/0.07$ during the first 30 epochs, while the inverse temperature for the non-parametric cross-attention is scheduled from 2.0 to 1.0. We employ the data augmentation method from DINO [8] (*e.g.*, color jittering of brightness, contrast, saturation and hue, Gaussian blur and solarization) with preceding random crops and resizing (to 224×224) and make them asymmetric. The exact settings of augmentation are provided in the next section.

D.3 Data Augmentation

The augmentation settings in FLSL are based on the augmentation pipeline of DINO [8] with one key modification: the random cropping operation is made asymmetric for the teacher and student networks. In our approach, we begin by sampling two random crops from the input image using a large ratio (*e.g.*, $0.8 \sim 1.0$) at the same location but with different pixel treatments. From each of the crops, we further sample a smaller crop using a ratio of (*e.g.*, $0.5 \sim 1.0$). The smaller crops are then assigned to the student network, while the larger crop are passed to the teacher network. This asymmetry ensures that the queries from the student exist within the teacher's view. Conversely, using symmetric random cropping for both networks adversely affects training performance and leads to collapse. Details of the data augmentation pipeline are listed below. The operations are performed sequentially to produce each view.

- For *Teacher network*, random cropping an area uniformly sampled with a size ratio between 0.8 to 1.0, followed by resizing to 224^2. `transforms.RandomResizedCrop(224, scale=(0.8, 0.1))` in PyTorch.

- For *Student network*, random cropping the crops from teacher network with an area uniformly sampled with a size ratio between 0.5 to 1.0, followed by resizing to 224^2. This results in an effective scale ratio of $(0.4, 1.0)$. `transforms.RandomResizedCrop(224, scale=(0.5, 1.0))` in PyTorch.

- Color jittering of brightness, contrast, saturation and hue, with a probability of 0.8. `ColorJitter(0.4, 0.4, 0.2, 0.1)` in PyTorch.

- Grayscale with a probability of 0.2. `transforms.RandomGrayscale(p=0.2)` in PyTorch.

- Gaussian blur with a probability of 0.5 and uniform random radius from 0.1 to 2.0.

- Solarization with a probability of 0.2.

- Color normalization with mean $(0.485, 0.456, 0.406)$ and standard deviation $(0.229, 0.224, 0.225)$.

D.4 PyTorch Pseudocode of FLSL

Algorithm 1 FLSL PyTorch Pseudo-code

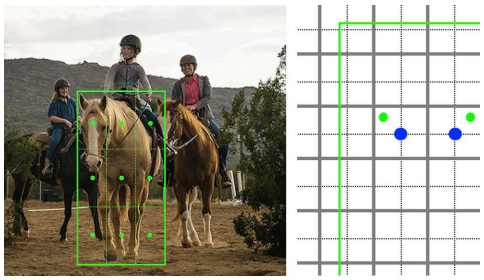
```
# fs, ft:  student and teacher transformer branches
# sa, ca:  self-attention and cross-attention head
# fc:  fully-connected layer
# tp_s, tp_t:  student and teacher inverse temperatures
# a, g, r:  coefficient for the three loss terms
# l:  network momentum rates
ft.params = fs.params
for x in Loader:# load a minibatch x with B samples
     # random augmentation
     x1, x2 = transforms_t(x)
     x1_s, x2_s = transforms_s(x1, x2)
     s1, s2 = fs(x1_s), fs(x2_s)# [B, N, D]
     t1, t2 = ft(x1), ft(x2)# [B, N, D]

     loss = 0.5 * M(s1, t2) + 0.5 * M(s2, t1)
     loss.backward()# back-propagation

     # student and teacher updates
     updates(fs)# SGD
     ft.params = l*ft.params + (1 − l)*fs.params

def H(s, t):
     # s, t:[B, N, D]
     zs, zt = fc(s), fc(t) # [B, N, K]
     ps, pt = softmax(zs/tp_s, dim=-1), softmax(zt/tp_t, dim=-1)
     ps_b = ps.sum(dim=-2).mean(dim=-1)
     return - (pt * log(ps)).sum(dim=-1).mean(),  ps_b*log(ps_b)

def M(s, t):
     t.detach()# stop gradient
     s0 = s.normalize(dim=-1)
     s = sa(s)
     t = ca(s, t, t)
     s0_a = s.normalize(dim=-1)
     h1, h2 = H(s, t)
     ds = ((s0 − s0_a) * (s0 − s_a)).sum(dim=-1).mean()
     return a * ds + g * h1 + r * h2
```

E Protocol for hyperparameter tuning

As discussed in the main paper, we need a protocol to evaluate the quality of the learned dense features during the FLSL training for hyperparameter tuning. However, standard evaluation protocols, such as k-NN classifier or linear probing are not suitable. We therefore propose a bounding box-aligned k-NN classification by leveraging the bounding box information provided by ILSVRC [44].

As shown in Figure 4(a), we partition the bounding box into $s \times s$ grids and find the coordinates of the center for each grid (the green dots). We then locate the s^2 features in the feature map, $\hat{\boldsymbol{Z}}$, from the nearest neighbor as shown in Figure 4(b), and store them into the memory bank with label information. For images with multiple bounding box annotations, we pick the largest one. An image is considered correctly classified as long as there is one of the s^2 features matching its true category with the prediction. We set $s = 3$ for our training and inflate the number of the nearest neighbors k by a scale factor c_s as the memory bank increases 9 times. We set $k = 20$ and $c_s = 7$ for the best performance.

We present the evaluation results of the bounding box-aligned k-NN of FLSL with the standard instance-level k-NN of other methods in Table 9. These results provide insights into the global and local semantic coherence of the learned representations. As the bounding box-aligned k-NN results in representations with less noise, we mark our results with $(*)$ symbol to indicate a **biased** comparison. Note that FLSL is designed for dense prediction tasks and not for instance-level image classification.



(a) (b)

Figure 4: The alignment between bounding box grid centers and the feature centers. We first construct a 3×3 grid from the bounding box and locate the grid centers. As shown in (a), the 9 grid center points are marked in green. Given the patch size (*e.g.*, 16×16) for each grid center, we then locate the patch with its center closest to the grid center, as shown in (b).

Method	Arch.	#params	#epochs	im/s	k-NN
Supervised	RN-50	23M	300	1237	79.3
SOTA SSL methods with Big CNNs					
SwAV	RN50w5	586	800	76	67.1
BYOL	RN200w2	250	1000	123	73.9
SimCLR-v2	RN152w3+SK	794	1000	76	73.1
Supervised	ViT-S/16	21M	300	1007	79.8
BYOL	ViT-S/16	21M	600	1007	66.6
MoCov2	ViT-S/16	21M	600	1007	64.4
MoCov3	ViT-S/16	21M	1200	1007	66.5
SwAV	ViT-S/16	21M	2400	1007	66.3
iBOT	ViT-S/16	21M	3200	1007	75.2
DINO	ViT-S/16	21M	3200	1007	74.5
FLSL	ViT-S/16	21M	1600	1007	76.7*
Comparison across transformer variants					
DINO	ViT-B/16	85M	1200	312	76.1
MoCov3	ViT-B/16	85M	1200	312	69.7
EsViT	Swin-S	49M	600	467	76.8
EsViT	Swin-B	87M	600	297	77.7
iBOT	Swin-T	28M	1200	726	75.3
iBOT	ViT-B/16	85M	1600	312	77.1
iBOT	ViT-L/16	307M	1000	102	78.0
DINO	ViT-B/8	85M	1200	63	77.4
DINO	ViT-S/8	21M	3200	180	78.3
EsViT	Swin-S/W=14	49M	600	383	77.3
EsViT	Swin-B/W=14	87M	600	254	78.3
iBOT	Swin-T/W=14	28M	1200	593	76.2
FLSL	ViT-B/16	85M	600	312	77.8*

Table 9: K-NN CLASSIFICATION ON IMAGENET

This Bbox-aligned *k*-NN classification is employed only for hyperparameter tuning and ablation study of the FLSL pipeline.

F Transfer learning settings

MS-COCO setup We evaluate the performance of the pretrained models on the MS-COCO object detection and instance segmentation tasks with different two-staged frameworks. For ViT-S/16 and ViT-S/8 with Mask R-CNN [24] and FPN [34], we employ multi-scale training following [6] and resize the image to ensure the short side falls within the range of 480 to 800 pixels, while ensuring the long side does not exceed $1,333$ pixels. For a fair comparison, we primarily adhere to the training setting utilized in [60]. Specifically, we employ the AdamW optimizer with a batch size of 16. Learning rate is linearly warmed up for the first $1,000$ iterations to reach $5e-5$ and subsequently decayed at step 8 and 11. Models are trained under 1x schedule. For ViT-B/16 with Mask R-CNN

and a simple FPN, we follow the training methodology outlined in Li et al. (2022) [33]. Specifically, the input images are resized to $1,024 \times 1,024$ and augmented with large-scale color jitter ranging from 0.1 to 2.0. The model is fine-tuned for 100 epochs using the AdamW optimizer with a weight decay of 0.1. To adjust the learning rate, we employ a step-wise decay strategy. During the training, the base learning rate is set to 0.0001, which is gradually increased from 0.0 to the base rate for the first 250 iterations as a warm-up phase. Additionally, we apply a layer-wise learning rate decay of 0.7.

UAVDT setup The UAVDT dataset contains $23,258$ images for training and $15,069$ images for test. The resolution of the images is about $1,080 \times 540$ pixels. The dataset is acquired with a UAV platform at a number of locations in urban areas. The categories of the annotated objects are car, bus, and truck. The training configuration is adapted from the original setting in [59]. The input size is rescaled to $1,072 \times 528$. The model is trained under 1x schedule. We adopt SGD optimizer with 0.9 momentum, 0.0001 weight decay and a batch size of 16. The base learning rate sets to 0.0005 with a linear warm-up for the first 300 iterations. The learning rate decreases at the 8th epoch.

G Ablation study

G.1 Impact of batch size

We study the impact of the batch size on the features extracted by FLSL. Table 10 shows that FLSL can achieve high performance with small batch sizes. Unlike the instance-level SSL methods that tend to focus on foreground contents (*e.g.*, objects), FLSL considers all the semantics in an image, *i.e.*, all the features $\boldsymbol{z}$s find their own cluster representatives $\hat{\boldsymbol{z}}$s through the self-attention (*mean-shift*) update. This enriches feature diversity and improves the variance of a mini-batch and benefits the training with small batch sizes.

Batch size	64	128	256	512	1024	2048
k-NN top-1	66.1	69.8	71.7	72.4	72.4	71.9

Table 10: IMPACT OF BATCH SIZE

G.2 Impact of random pooling

In FLSL, contrasting among dense features can be computationally expensive, *i.e.*, $14^2 = 196$ representations to be considered in the objective. Therefore, we apply a random pooling to the queries from the last ViT layer and study the impact of different window sizes of the random pooling.

Window size	2×2	4×4
k-NN top-1	72.4	71.1

Table 11: IMPACT OF RANDOM POOLING

G.3 Impact of the number of centroids K

We formulate FLSL as an explicit clustering problem. Therefore, the output dimension of the last fully-connected layer is equal to the number of centroids K. As shown in Table 12, FLSL enjoys a smaller output dimension compared to its instance-level counterpart, DINO ($K = 65,536$) [8]. This is mainly due to the higher variance of features in an image than that of a feature cluster. Take ImageNet for instance, the content of an image may range from a single object and stuff to a melange of them from different categories. This requires a large number of centroids to cover the image distribution. While for a semantic cluster, it tends to contain features of high correlation, *e.g.*, features of similar texture, or multiple adjacent objects from the same category, hence requires less centroids to cover its distribution. From the experiment, we find that a large number of centriods improves the performance, but is detrimental and costly when being too large. We pick $K = 4,096$ for all our experiments as it strikes a good balance between performance and cost-effectiveness.

K	1024	2048	4096	8192	16384
k-NN top-1	68.1	72.1	72.4	72.5	72.1

Table 12: IMPACT OF NUMBER OF CENTROIDS K

image $\upsilon/\eta = 0.1$ $\upsilon/\eta = 0.2$ $\upsilon/\eta = 0.3$ $\upsilon/\eta = 1.$ image $\upsilon/\eta = 0.1$ $\upsilon/\eta = 0.2$ $\upsilon/\eta = 0.3$ $\upsilon/\eta = 1.$

Figure 5: Impact of the ratio υ/η on local semantic consistency with the FLSL-learned representations. The figure presents a visualization of the aggregated attention scores (AAS) map. As the ratio υ/η increases, the attention for each query becomes more focused, specifically attending to regions of closer proximity, resulting in more cluttered and smaller dark regions in the AAS map.

G.4 Ablation on the FLSL objective function

The FLSL objective contains three components: (1) similarity between ℓ_2-normalized $\boldsymbol{z}$ (features) and $\hat{\boldsymbol{z}}$ (modes), (2) cross-entropy of the probabilities of an augmented pair $H(p(\hat{\boldsymbol{z}}^+), p(\hat{\boldsymbol{z}}))$, and (3) the non-empty constraint $D_{\text{KL}}\left(\bar{\boldsymbol{p}} \| \boldsymbol{\pi}\right)$:

$$\min \frac{1}{N'} \sum_{\boldsymbol{Z} \in \mathcal{Z}} \sum_{\boldsymbol{z} \in \boldsymbol{Z}} \upsilon \|\boldsymbol{z} - \hat{\boldsymbol{z}}\|_{\mathcal{F}}^2 + \eta \sum_{\boldsymbol{z} \in \boldsymbol{Z}} \mathrm{H}(\boldsymbol{p}(\hat{\boldsymbol{z}}^+), \boldsymbol{p}(\hat{\boldsymbol{z}})) + \gamma D_{\text{KL}}\left(\bar{\boldsymbol{p}} \| \boldsymbol{\pi}\right), \tag{31}$$
$$\text{with } \hat{\boldsymbol{z}} = \mathrm{SA}(\boldsymbol{z}, \boldsymbol{Z}, \boldsymbol{Z}),\ \hat{\boldsymbol{z}}^+ = \mathrm{CA}(\boldsymbol{z}, \boldsymbol{Z}^+, \boldsymbol{Z}^+).$$

Sinkhorn	η	γ	$\upsilon = 0.0$	$\upsilon = 0.1$	$\upsilon = 0.2$	$\upsilon = 0.3$	$\sim$	$\upsilon = 1.0$
✓	1.0	1.0	0.1	68.7	70.7	71.2	$\sim$	65.1
×	1.0	1.0	-	-	-	66.6	-	-
×	1.0	5.0	-	-	-	72.4	-	-

Table 13: IMPACT OF THE COEFFICIENTS IN THE FLSL OBJECTIVE.

It is computationally expensive to optimally determine the values of more than two coefficients by performing grid search, especially when the ratios among them are large. We tackle this problem by first fixing $\eta = 1$ and setting $\gamma = 1$ along with the Sinkhorn normalization [16] to perform a grid search on the value of υ with the empirical base condition $\upsilon \leq 1$ and $\gamma \geq 1$ [60, 1]. With the fixed υ, we then perform another grid search on γ without the Sinkhorn normalization. We implement Sinkhorn normalization [16] as the softmax operation along the batch dimension. Table 5 summerizes the score of *k*-NN evaluation using different coefficient settings. We also visualize the impact of different ratios of the first and second level clustering υ/η of the FLSL objective in Figure 5 by visualizing the aggregated attention score (AAS) map. As the ratio increases, the AAS map shifts from being clear and bright to becoming cluttered and dark. This change occurs because the self-attention for each query becomes more focused, attending to a smaller neighborhood. A smaller ratio leads to larger clusters, which aggregate more attention scores in the region, resulting in a brighter map, particularly in the background. Conversely, a large ratio leads to small, cluttered clusters with fewer attention scores aggregated, resulting in a darker map. A smaller ratio may smooth out small details, while a larger ratio causes the model to focus excessively on local features. From the results in Table 13, a ratio of 0.3 strikes a good balance in between.