Clusformer: A Transformer based Clustering Approach to Unsupervised Large-scale Face and Visual Landmark Recognition

Xuan-Bac Nguyen[1], Duc Toan Bui[1], Chi Nhan Duong[2], Tien D. Bui[2], Khoa Luu[3]
[1] VinAI Research, Vietnam [2] Concordia University, Canada [3] University of Arkansas, USA
[1]{v.bacnx2, v.toanbd1}@vinai.io, [2]{dcnhan@ieee.org, bui@encs.concordia.ca}, [3]khoaluu@uark.edu

Abstract

The research in automatic unsupervised visual clustering has received considerable attention over the last couple years. It aims at explaining distributions of unlabeled visual images by clustering them via a parameterized model of appearance. Graph Convolutional Neural Networks (GCN) have recently been one of the most popular clustering methods. However, it has reached some limitations. Firstly, it is quite sensitive to hard or noisy samples. Secondly, it is hard to investigate with various deep network models due to its computational training time. Finally, it is hard to design an end-to-end training model between the deep feature extraction and GCN clustering modeling. This work therefore presents the Clusformer, a simple but new perspective of Transformer based approach, to automatic visual clustering via its unsupervised attention mechanism. The proposed method is able to robustly deal with noisy or hard samples. It is also flexible and effective to collaborate with different deep network models with various model sizes in an end-to-end framework. The proposed method is evaluated on two popular large-scale visual databases, i.e. Google Landmark and MS-Celeb-1M face database, and outperforms prior unsupervised clustering methods. Code will be available at https://github.com/VinAIResearch/Clusformer

1. Introduction

The research in automatic unsupervised visual clustering, e.g. human faces or landmark photos, has gained considerable prominence lately thanks to the nature of huge amount of available *unlabeled* data and the demand of consistent visual recognition algorithms across various challenging conditions. Indeed, stand-alone visual recognition algorithms, e.g. Face Recognition [36] or Visual Landmark Recognition [38], are important in practical applications where there is significant difference between probe and gallery visual photos [23]. In Face Recognition, the algorithms of face recognition in a supervised manner have now become mature. Compared to the state-of-the-art (SOTA) results of supervised Face Recognition algorithms [7, 15], the number of studies in face clustering are still limited. The solutions are still not as good as supervised methods.

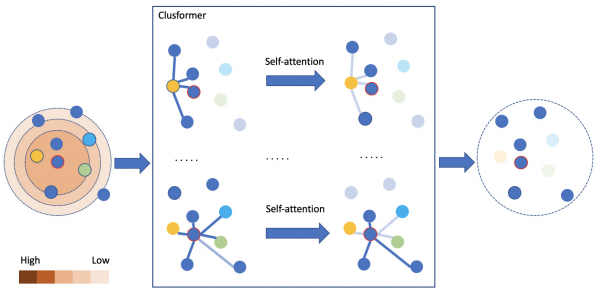

Figure 1. The proposed Clusformer uses the self-attention mechanism to detect the hard, noisy samples in a cluster, while prior methods use GCNs that are unable to address this problem completely.

Many complex factors could affect the appearance of a visual photo, e.g. illumination, poses, occlusions, in real-world scenarios. Providing tolerance to these factors is the main challenge for accurate visual clustering methods. Among these factors, lacking of robust features is often the most important factor to deal with. Visual data is usually easy to collect but costly to annotate. Therefore, Graph Convolutional Networks (GCNs) have become one of the most popular methods to tackle visual clustering in an unsupervised manner. However, recent visual clustering methods, e.g face clustering [14, 27, 18, 40, 39], still have some limitations, e.g. accurate clustering, algorithm complexity or computational time.

1.1. Contributions of This Work

This work presents the new Clusformer, a simple but novel perspective of Transformer based approach, to automatically cluster visual samples in an unsupervised manner. The method is able to robustly deal with noisy and hard samples thanks to its effective self-attention mechanism. To

the best of our knowledge, it is one of the first work to utilize the self-attention mechanism in Transformer to tackle visual clustering problems effectively. The contributions of this work are therefore three-fold. Firstly, a new Transformer based clustering architecture is introduced in the context of top-down clustering large-scale unsupervised visual databases. Secondly, new Visual Grammar and Cosine Distance Encoding (CDE) modeling mechanisms are introduced to efficiently incorporate into the Clusformer framework to solve the visual clustering problems. Finally, the proposed approach consistently achieves the state-of-the-art (SOTA) results compared against the recent clustering methods [40, 39] on two standard visual benchmarks, i.e. Google Landmark and MS-Celeb-1M face database.

2. Related Work

This section briefly reviews recent methods for face and visual landmark clustering. They can be divided into three main categories, including unsupervised, semi-supervised and supervised visual clustering methods.

2.1. Unsupervised Visual Clustering

These methods usually compute empirical density and designate clusters as dense regions in a data space such as K-Means [21] and spectral clustering [32]. Otto et al. [27] introduced an approximate rank order metric for clustering millions of faces by identity. Ankerst et al. [1] developed a similar concepts and addresses the ordering of data points. Chen et al. [4] proposed an unsupervised hashing method named Anchor-based Probability Hashing to preserve the similarities by exploiting the distribution of data points.

2.2. Semi-Supervised Face Clustering

These methods aim to leverage large-scale unlabeled data, given a handful of labeled data. Roli at al. [30] proposed a self-training strategy that employs Principal Component Analysis (PCA) to leverage labels and unlabelled data with an initial classifier. Zhao et al. [43] employed Linear Discriminant Analysis (LDA) as the classifier to infer labels. Zhan et al. [42] introduced a Consensus-Driven Propagation (CDP) to exploit massive unlabeled data for improving large-scale face clustering.

2.3. Supervised Face Clustering

These methods rely on supervised information to improve performance gains. Lin et al. [19] proposed to exploit local structures of deep features by introducing minimal covering spheres of neighbourhoods to improve similarity measure. GCNs [18] extend Convolutional Neural Networks (CNNs) to process graph structured data. Wang et al. [37] further improved the linkage prediction by leveraging GCNs to capture graph context. Hyeonwoo et al. [25] introduced an attentive local feature descriptor suitable for large-scale image retrieval. Teichmann [34] proposed a regional aggregated selective match kernel to effectively combine information from detected regions into an improved holistic image representation for visual landmark clustering. Cao et al. [2] presented an unifying deep local and global features for image retrieval. Jerome et al. [29] proposed method to directly optimize the global mAP by leveraging recent advances in listwise loss formulations.

The proposed method adopts the idea of **supervised clustering**, it differs from two key aspects: (1) we introduce the **Clusformer**, a simple but new perspective of Transformer based approach, to automatic unsupervised visual clustering via its efficient unsupervised attention mechanism. (2) our method is able to **robustly deal with noisy or hard samples**. It's also **flexible and effective** to collaborate with different deep network models with various model sizes and be able to train and optimize with these deep network within an end to end network.

3. Background

This section reviews Graph Convolutional Neural Networks and their limitations. Then we briefly review the Transformer method and its self-attention mechanisms.

3.1. Graph Convolutional Neural Networks (GCNs)

GCNs are originally invented in the field of spectral graph theory [18] and graph signal processing [26]. They are also related to the spectral graph theory [18]. GCNs use the same ideas in spectral graph theory to design the parameterized filters to fit in CNNs. GCNs have showed as one of the most effectiveness methods in dealing with complex cluster patterns. However, they are usually highly computational and require high level of memory consumption. In addition, they are quite sensitive to hard or noisy samples. In [39], the confidence of a vertex v_i is measured in the affinity graph as shown in Eq. (1).

$$c_i = \frac{1}{|\mathbf{N}_i|} \sum_{v_j \in \mathbf{N}_i} (1_{y_j = y_i} - 1_{y_j \neq y_i}) \cdot a_{i,j}, \qquad (1)$$

where 1 denotes the identity function that defines v_i and v_j are same class or not. c_i is the confidence score that the neighbors $\mathbf{N}_i$ are close or from the same class as v_i. The affinity $a_{i,j}$ between two vertices v_i and v_j is computed by $a_{i,j} = 1 - s_{i,j}$ where $s_{i,j}$ is the cosine similarity score. When there are more noisy samples around v_i, the value of $a_{i,j}$ is much higher and c_i is penalized by $-a_{i,j}$ that leads to a lower value. Similarly, when there are many hard samples in the same class with v_i, but they are far away from v_i, the value of $a_{i,j}$ is much smaller, then the confidence score s_i would be slightly increased. In these two scenarios that we mentioned, GCNs are unable to solve them completely.

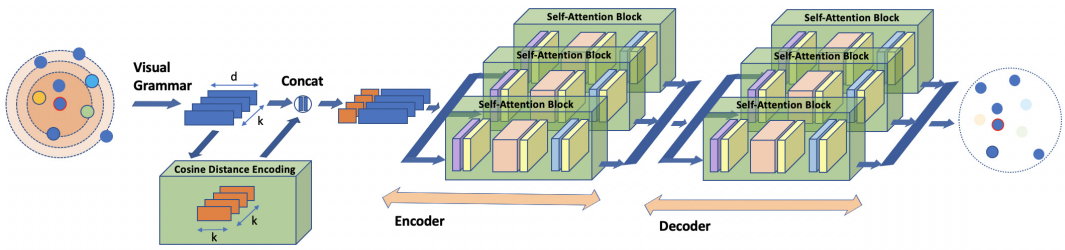

Figure 2. Clusformer Architecture Overview

In addition, the combination between a deep network model and a GCNs model is still hand-crafted and executed in an alternative manner. Their efficiency also suffers from the requirement of numerous highly computational training time. Indeed, recent GCN-based clustering methods [40, 39] have been introduced in an alternative manner.

3.2. Transformer

Attention, a mechanism that allows a Neural Network to focus on some particular regions of an input when making a prediction, has become a key component in most recent successful deep architectures. When combined with CNNs, attention mechanisms have achieved very high accuracy in numerous Natural Language Processing (NLP) applications. Recently, the Transformer [35], an encoder-decoder architecture based merely on attention mechanisms, has achieved the SOTA results on sequence-to-sequence tasks. With the multi-head attention, Transformer can focus on different positions in an input sequence and results in a powerful deep network that outperforms prior approaches. Following the success of Transformer, many Transformer-based architectures have been presented recently to obtain SOTA results in various NLP problems, e.g. GPT [28], BERT [9], Transformer-XL [6] , XLNet [41] and among others [20, 24, 13, 5]. In this work, we aim to present a simple but new perspective of Transformer in visual clustering thanks to their effective self-attention mechanism.

4. The Proposed *Clusformer* Approach

This section firstly overviews the proposed framework in Subsection 4.1. Then we detail how to initialize visual clusters from large-scale datasets in Subsection 4.2. Finally, the proposed Clusformer is presented in Subsection 4.3.

4.1. Framework Overview

There are two training visual datasets in this work, i.e. a visual dataset with full annotation $\mathbf{D}_L = (\mathbf{X}_L, y_L)$, and a large-scale visual dataset *without* any annotation $\mathbf{D}_U = (\mathbf{X}_U, \varnothing)$, where $\varnothing$ denotes the label is not available. In practice, $\mathbf{D}_U$ is much greater than $\mathbf{D}_L$ in terms of both the number of training samples and the number of subject identities. Obviously, it requires a huge amount of efforts to *manually* annotate labels for $\mathbf{D}_U$ and as such that is impractical in the real world.

A CNNs model $\mathcal{M}$ is presented for visual classification. One of the efficient approaches to improve the accuracy of the classifier $\mathcal{M}$ is to maximize the usage of both the labeled dataset $\mathbf{D}_L$ and the unlabeled dataset $\mathbf{D}_U$. Thus, the deep model $\mathcal{M}$ is firstly trained using the labeled training set $\mathbf{D}_L$ in a regular supervised manner. Then, this classifier $\mathcal{M}$ is used to bootstrap training samples in the large-scale unlabeled dataset $\mathbf{D}_U$. A high performance clustering method is used to automatically label the dataset $\mathbf{D}_U$. This clean large-scale dataset will be then used to retrain the visual classifier model $\mathcal{M}$. Introducing a high accuracy of an automated clustering method to improve the classifier $\mathcal{M}$ is the goal of this work. We introduce a new simple but efficient Clusformer method to cluster the large-scale unlabeled dataset $\mathbf{D}_U$. Figure 2 overviews the proposed framework.

It is important to notice that these initial clusters $\mathbf{C}_i$ may contain mislabeled samples. Clusformer as shown in Subsection 4.3 are introduced to improve the accuracy in these initial clusters $\mathbf{C}_i$ by detecting hard and noisy samples thanks to the self-attention mechanism. In addition, the deep model $\mathcal{M}$ can be replaced by any recent deep CNNs that allows high-performance. It is the same for the initial clustering method. These two models are considerable and well selected but not the main focus in this work.

4.2. Visual Cluster Initialization

Given a visual classifier $\mathcal{M}$, an input image $x_i \in \mathbf{X}^{h \times w \times 3}$ is fed to the model $\mathcal{M}$ to extract visual features $\mathbf{f}_i = \mathcal{M}(x_i)$, where h, w denote the height and width of the input image. From now on, for further convenience, the term of features is used to represent the visual images and vice versa. For each sample in the dataset, the cosine similarity based k-nearest neighbors $\mathcal{K}$ is used to cluster input samples based on their similarity scores. They will be formed as a cluster $\mathbf{C}_i$ that has $\mathbf{f}_i$ as the center.

$$\mathbf{C}_i = \mathcal{K}(\mathbf{f}_i, \mathbf{F}, k) \tag{2}$$

where $\mathbf{F} = \mathcal{M}(\mathbf{X})$ is a set of features extracted from $\mathbf{X}$. k denotes the number of nearest neighbors. Alternatively, we construct a cluster dataset denoted as $\mathbf{C} = \{\mathbf{C}_i\}_{i=1}^{N}$. This dataset will be mainly used in the next sections.

Algorithm 1 Visual Grammar: Constructing a visual sequence

Input: Input image x_i, Visual classifier $\mathcal{M}$, data set $\mathbf{X}$, length of visual sequence k, and step size Δ_r
Output: Visual Sequence $\mathbf{S}_i$
$\mathbf{S}_i = \emptyset$; $\mathbf{f}_i = \mathcal{M}(x_i)$
$radius = 0$
while $|\mathbf{S}_i| \leq k$ **do**
  $S = \{x_j \in \mathbf{X} | radius \leq s_{ij} \leq radius + \Delta_r\}$ where $s_{i,j} = \text{dist}(\mathbf{f}_i, \mathcal{M}(x_j))$
  if $S = \emptyset$ **then**
    break
  end if
  $ind \leftarrow (\text{argsort}(S))_v := |\{u \in \{1, ..., |S|\} | x_u \in S, s_{i,u} \leq s_{i,v}\}|$
  for all u in ind **do**
    $\mathbf{S}_i = \mathbf{S}_i + [\mathcal{M}(x_u)], x_u \in S$
  end for
  $radius = radius + \Delta_r$
end while
return $\mathbf{S}_i$

4.3. Self-Attention Clustering Approach

4.3.1 Visual Grammars - From Clusters to Sequences

In GCNs [40, 39], the cluster dataset $\mathbf{C}$ is used to build the affinity graphs. Each cluster $\mathbf{C}_i$ is represented represents as a graph where each image is a vertex and the edge between vertexes is expressed by their similarity scores. However, these data structures are unable to be fed into the Transformer directly because the input has to be represented as a *sequence*. Intuitively, in NLP, the order of words is arranged by a pre-defined rule called grammars. Each word in different positions might have a different meaning. Thus, it raises a question: *How to define a function that constructs a visual sequence from a given cluster? What is the appropriate order of visual samples?*.

We present a simple but effective visual grammar $\mathcal{G}$ to formulate a visual sequence $\mathbf{S}_i$ from an unordered cluster.

$$\mathbf{S}_i = \mathcal{G}(\mathbf{C}_i) \tag{3}$$

The cluster $\mathbf{C}_i$ has a center $\mathbf{f}_i$, where $\mathbf{f}_j$ is denoted as the j^{th} neighbor of $\mathbf{f}_i$ in this cluster $1 \leq j \leq k$. $s_{i,j}$ is the cosine similarity score between $\mathbf{f}_i$ and $\mathbf{f}_j$. The similarity scores of all the neighbors are measured to the center forming a vector, i.e. $\{s_{i,j}\}_{j=1}^k$. Next, the order of element in this vector is sorted in descending order and re-arranges visual samples in $\mathbf{C}_i$ following this form. The visual initialization and construction of visual grammar $\mathcal{G}$ are detailed in Algorithm 1 and illustrated in Figure 3.

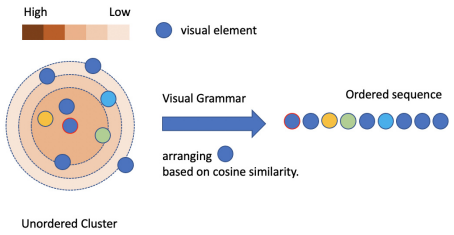

Figure 3. The visual grammar is designed to convert un-ordered cluster to ordered sequence. The element which has higher cosine similarity score will be more close to the center while the element with lower score will be far from the center in the sequence.

4.3.2 Visual Cluster Encoder

Self-attention. In section 4.3.1, a visual sequence $\mathbf{S}_i$ is obtained from a cluster $\mathbf{C}_i$. Each visual sample $f_i \in \mathbf{S}_i$ has a feature of $1 \times d$, $\mathbf{S}_i \in \mathbb{R}^{k \times d}$. $\mathbf{S}_i$ will be projected into three super spaces called: *key*, *query* and *value*. We define three learnable matrixes: $\mathbf{W}^Q \in \mathbb{R}^{d \times d'}$, $\mathbf{W}^K \in \mathbb{R}^{d \times d'}$ and $\mathbf{W}^V \in \mathbb{R}^{d \times d'}$. The query, key and value features will be computed as follows,

$$\begin{aligned} \mathbf{Q} &= \mathbf{S}_i \mathbf{W}^Q, \mathbf{Q} \in \mathbb{R}^{k \times d'} \\ \mathbf{K} &= \mathbf{S}_i \mathbf{W}^K, \mathbf{K} \in \mathbb{R}^{k \times d'} \\ \mathbf{V} &= \mathbf{S}_i \mathbf{W}^V, \mathbf{V} \in \mathbb{R}^{k \times d'} \end{aligned} \tag{4}$$

In order to compute the relevancy between the i^{th} visual sample and the j^{th} visual sample in this sequence, the attention score is computed as in Eqn. (5).

$$r_{i,j} = \frac{e^{\frac{1}{\sqrt{d'}} \mathbf{Q}_i \mathbf{K}_j^T}}{\sum_{j=1}^k e^{\frac{1}{\sqrt{d'}} \mathbf{Q}_i \mathbf{K}_j^T}} \tag{5}$$

$\mathbf{Q}$ and $\mathbf{K}$ are used to construct the relationships between one visual sample to others in the sequence. The value $\mathbf{V}$ will be used to summarize all of them. The final output $\mathbf{Z}$ is the aggregation of $\mathbf{V}$ by the weighted attention score.

$$\mathbf{Z} = Att(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \{\mathbf{Z}_i\}_{i=1}^k \tag{6}$$

where $\mathbf{Z}_i = \sum_{j=1}^k r_{i,j} \cdot V_j$.

Multi-head attention. In the self-attention algorithm, each feature is projected into a different space by the learnable matrix weights. In multi-head attention, it is projected into several sub-spaces. It leads to the model capable of paying attention to different positions in the sequence. Let m be the number of sub-spaces. Each sub-space, the feature dimension of each space is $d' = \frac{d}{m}$. The output is linear transformation of concatenation of all attention outputs.

$$\mathbf{Z}_M = concat(\mathbf{Z}_{s,1}, \ldots, \mathbf{Z}_{s,m}) \cdot \mathbf{W}^M \tag{7}$$

where: $\mathbf{Z}_{s,i} = Att(\mathbf{Q}_{s,i}, \mathbf{K}_{s,i}, \mathbf{V}_{s,i}), 1 \leq i \leq m$ and $\mathbf{W}^M$ is an additional weight matrix. This output is passed to the

Point-wise Feed Forward Network (FFN) to formulate an encoder block.

4.3.3 Clusformer Decoder

Similar to the encoder block, the decoder block consists of self-attention modules and decoder-encoder attention modules in addition. Apart from the original decoder, it produces an output as a sequence in parallel while [35] generates output prediction one element at a time. Indeed, we define a learnable visual query matrix with a shape of $k \times d$. It is fed into the input of each attention layer directly.

4.3.4 Cosine Distance Encoding

The Clusformer encoder expects an input in the form of a sequence. Thus, visual sequence $\mathbf{S}_i$ from a cluster in section 4.3.2 will be constructed. This sequence has the dimension of $k \times d$, where k is the length of the sequence and d is the visual feature dimension. To compare against GCNs based methods, a cluster is represented as a graph with two parts, i.e. vertices and edges. A vertex is a single image with the $1 \times d$ feature in a disordered manner. An edge is the cosine similarity of one vertex to the rest of vertexes inside this sub-cluster. Both these components are showed as a matrix and fed into the GCN. Thus, both visual features and the cosine similarity distance are helpful for visual clustering.

Our visual sequence only carries the visual information while missing the pair-wise cosine similarity information. To address this problem, we propose a new encoding method named Cosine Distance Encoding (CDE). Our CDE has the following differences in comparing with the Positional Encoding (PE), widely used in NLP [35] or Computer vision [3]. Firstly, the order and position of words are essential parts of any language. They are considered as the sentence grammar. The same word in different positions will lead to different meanings. Previously, Recurrent Neural Networks or Long-Short Term Memory feed the word into the network in sequence. However, thanks to Transformer, all the words are passed through in parallel. Thus, it raises a problem that Transformer does not have any sense of each word and its position. The words in a sentence thus needs to be cooperated. For this reason, the encoding, i.e Positional Encoding, is presented.

However, in visual clustering, the visual sequence already follows a grammar: the element that is closer to the center will be put ahead in the sequence. Thus, our encoding does not mean to describe grammar in our sequence. In addition, in NLP, encoding information will be added into the word features by addition-wise operation. In our problem, the visual feature is normalized by L_2 normalization. Hence, adding some values to the vector would lead to shifting or expanding the cluster. Thus, we do not perform additional-wise. Instead, we use concatenation function. Let t is the position of an element in the sequence. $\mathbf{e}_t \in \mathbb{R}^{1\times k}$ is the corresponding CDE.

Figure 4. Illustration of MS-Celeb-1M dataset. Each row represents as an identity. The images with the green border belong to same identity while images with the red border are the hard samples. The first image of each row is the center image of a cluster.

$$\mathbf{e}_t = \{s_{t,i}\}_{i=1}^{k} \tag{8}$$

The feature of t^{th} element the sequence turns to:

$$\mathbf{f}_t = concat(\mathbf{f}_t, \mathbf{e}_t) \tag{9}$$

Finally, the input sequence fed into the Clusformer has shape of $k \times (d + k)$.

4.3.5 Objective and Loss Function

Objective. In Subsection 4.3.4, we discuss how to construct a visual sequence to feed to the Clusformer. In this section, we present its objective and the output sequence. The visual sequence $\mathbf{S}_i$ is constructed from the sub-cluster $\mathbf{C}_i$ formulated via a clustering algorithm. The neighbors of f_i are expected to have the same label as the center. However, in the wild conditions, it exists hard samples from various clusters or identities. Thus, $\mathbf{C}_i$ does not contain visual samples from one identity totally. The Clusformer is presented to detect these hard samples. The output sequence is a binary sequence y_i having the same length as $\mathcal{S}_i$. Let y_i^t be the value of t^{th} element of y_i. When $y_i^t = 1$, the t^{th} element in the sequence has same label as the center f_i and vice versa.
Loss function. Let $\hat{y}_i$ be the predicted output. Binary Cross Entropy loss is used to train the Clusformer.

$$\mathcal{L}_i(\hat{y}_i, y_i) = -\sum_{t=1}^{k}[\, y_i^t \times \log(\sigma(\hat{y}_i^t)) + (1-y_i^t) \times \log(1-\sigma(\hat{y}_i^t))\,] \tag{10}$$

where σ is the sigmoid function.

5. Experiments

5.1. Face Clustering and Recognition

Dataset and **Protocol**. MS-Celeb-1M [11] is a large-scale face recognition dataset crawling from the internet. The cleaned version consists of 5.8M images from 85K identities. All the images are pre-processed by aligning and

cropping to the size of 112×112. We randomly split the MS-Celeb-1M into the 10 parts. Each part contains 584K images of 8,500 identities approximately. There is no identity overlapped among them. We randomly select one part denoted $\mathbf{D}_L = (\mathbf{X_L}, y_L)$ for fully supervised training. The rest are used as unlabelled set, i.e. $\mathbf{D}_U = (\mathbf{X}_U, \varnothing)$.

In the first step, a deep visual model $\mathcal{M}$ is trained with $\mathbf{X}_U$. Then, a visual sequence dataset is created for the both datasets using Eqn. (2) followed by the visual grammar of Eqn. (3).

$$\begin{aligned} \mathbf{S}_L &= \{\mathbf{S}_i\}_{i=1}^{N_L} = \{\mathcal{G}(\mathcal{K}(\mathbf{f}_i, \mathbf{F}_L, k))\}_{i=1}^{N_L} \\ \mathbf{S}_U &= \{\mathbf{S}_j\}_{j=0}^{N_U} = \{\mathcal{G}(\mathcal{K}(\mathbf{f}_j, \mathbf{F}_U, k))\}_{j=1}^{N_U} \end{aligned} \tag{11}$$

We train Clusformer with $\mathbf{S}_L$ and employ $\mathbf{S}_U$ as the test set. Let y_p be the *pseudo label* of $\mathbf{D}_U$ obtained by Clusformer. We combine both $\mathbf{D}_L$ and $\mathbf{D}_U$ into one dataset $\mathbf{D}_P = \mathbf{D}_L \cap \mathbf{D}_U$ and conduct a new deep model for retraining.

Metrics. For the face clustering, to measure the similarity between two clusters with a set of points, we use Fowlkes Mallows Score (FMS). This score is computed by taking geometry mean of precision and recall. Thus, FMS F_B is also called Pairwise-Fscore as follows,

$$F_B = \frac{TP}{\sqrt{(TP + FP) \times (TP + FN)}} \tag{12}$$

where TP is number of point pairs in the same cluster in both ground truth and prediction. FP is number of point pairs in the same cluster in ground truth but not in prediction. FN is number of point pairs in the same cluster in prediction but not in ground truth. Besides Pairwise F-score, BCubed-Fscore denoted as F_B is also used for evaluation.

For face recognition, we follow MegaFace [17] protocol, one of the largest benchmark for face recognition. It contains a set of probe from FaceScrub with 3,530 images and 1M gallery images. We select top-1 identification hit rate as the evaluation metric.

5.2. Visual Landmark Recognition

Dataset and **Protocol**. Google Landmarks Dataset Version 2 (GLDv2) [38] is a largest dataset about visual landmark recognition and identification. The cleaned version includes 1.4M images of 85K landmarks and 800 hours of human annotation. The landmarks are collected from all the corners in the world with diversity categories. The dataset is extremely long-tail distribution, the number of image per class varies from 0 to 10,000. In comparison with face recognition, the GLDv2 is similar but much more challenging. We randomly split the dataset into 3 parts. Each part contains 28K landmarks. There is no overlap between them. We pick one part for training the deep visual model and Clusformer. The rest are used for the testing. The training and inference produces are as in Section 5.1.

Figure 5. Different kinds of landmark on the visual landmark dataset. The images in each row with green border belong to same landmark while images with red border are the noisy/hard samples. The first image of each row is the center image of a cluster.

5.3. Implementation Details

Our framework is implemented in Pytorch and running on the desktop equipped with AMD EPYC 7742 64-Core Processor chipset. To speed up the training step, we utilize distributed training where each GPU is considered as a single process. We employ Resnet50 [12] as the visual model. We drop the last pooling and fully connected layers. Instead, a linear layer is adopted following the last convolution layers of model to form an embedding size of 256. To train large-scale and long-tailed datasets, [8] is employed for the loss function. The image size is set to 112×112 pixels for face dataset and 224×224 pixels for landmark dataset. We train the model with 24 epochs from scratch. The loss is optimized with the SGD optimizer. The weight decay and the momentum are 0.001 and 0.9, respectively. The batch size is 512. The initialized learning is set to 0.1 and reduced 10 times at epoch 12, 16 and 18.

Clusformer Training. We select the number of the closest neighbors $k = 64$ to build cluster datasets for both face and visual landmark datasets. Thus, the visual sequence is the length of 64. Dealing with a large-scale dataset, we use KNN based GPU algorithm [16] to speed up. The time consumption to find all the neighbors of 4.8M images is about 2 minutes approximately. Our Clusformer is designed to have 6 encoder and decoder blocks. The number of multi-head attention is 8. The model is trained in 50 epochs. AdamW optimizer [22] is used to optimize the loss. The learning rate is set to 0.0001 at the beginning and reduced linearly during the training.

5.4. Experiment Results

Face Clustering. Table 1 shows the clustering performance on MS-Celeb-1M dataset. We aim to evaluate the algorithm in different number of unlabelled data. Thus, there are 5 scenarios with different numbers of samples of the test set: 584K, 1.7M, 2.89M, 4.05M and 5.21M respectively. We compare with several methods including deep learning based [40, 39, 42], and typical [21, 31, 33, 10, 27] approaches. The running time in Table 1 is reported by doing inference on the first part 584K images. It is not surprised that the deep learning based methods outperform the typical

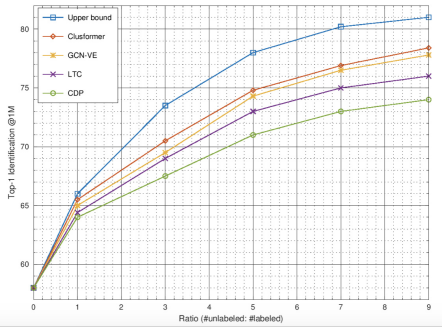

Figure 6. Identification Performance in Megaface Protocol

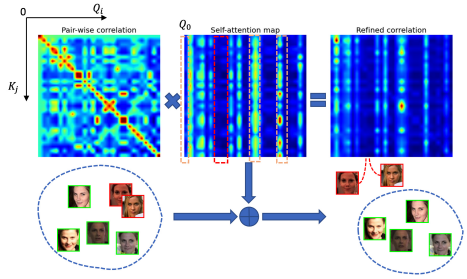

Figure 7. Self-attention visualization

one. It seems that KNN gives the best performance among typical methods. However, during testing, we assume that we already know exactly number of cluster in the test set. Thus, the performance of KNN is considered as the best performance that this algorithm is able to reach to. In practice, we do not know exactly number of clusters. Therefore, the performance would be a lower number. In comparison with deep learning approaches, we compare to the GCN based methods: CDP [42], L-GCN [37], LTC [40], GCN-V [39] and GCN-V + GCN-E [39]. Among these methods, CDP is the fastest method with 2.3 minutes running time. However the performance is not significant better than K-means. Even with a small amount of unlabelled data, CDP is lower than K-means by 3%-4% on both F_P and F_B metrics. When the test set increases, the performance margin is much more smaller. These results show that CDP is not stable. There is a remarkable jump of performance from GCN-V method compared to CDP. Both F_P and F_B scores are improved impressively. However, there is still a disadvantage remaining. The inference time is increased twice to 4.5 minutes for yeiling results of 584K samples. Besides, they also proposed a second network named GCN-E that takes output of GCN-V as the input and being trained to refine the final results. The final results increase a little bit, however the inference time is about 2.5 times longer.

In Table 1, it is clear to see that, Clusformer is better than the best of GCN based methods in term of accuracy and running time. Overall, our method performs better in all the scenarios. When there are more unlabeled samples, it still perform the best. It proves the stability of Clusformer. In addition, Clusformer is also faster than CDP in the inference step. In conclusion, Clusformer is fast and accurate.

Face Recognition. The performance of model retrained with combination of $\mathbf{D}_P = \mathbf{D}_L \cap \mathbf{D}_U$ is reported in Figure 6. The ratio of unlabeled and labeled dataset is denoted as: $r_{U/L} = \frac{N_U}{N_L}$ We experiment different values of $r_{U/L}$ to see how good deep visual model can be improved when N_U increases. In Figure 6, the upper bound is the curve that the model can reach to. It means that, in the assumption of Clusformer returns the perfect results, all the samples are assigned a pseudo-label correctly. The upper bound is the highest threshold the models are able to achieve. General speaking, Clusformer outperforms GCN based methods.

Visual Landmark Clustering This section analyzes and compares performance of Clusformer on visual landmark clustering. We experiment the GCN based methods: GCN-V, GCN-VE and L-GCN for comparisons. To be fair, we do hyper parameters tuning for these methods so that they get the best results on this dataset. It is interesting that GCN-VE is lower than GCN-V and L-GCN that is completely contrary to the face recognition dataset. In contrast, Clusformer still shows its robustness by achieving 19.32% and 40.63% of F_P and F_B correspondingly. Compared to L-GCN (14.08% F_P and 36.35% F_B), Clusformer achieves the best performance in this problem as shown in Table 2.

5.5. Ablation Study

5.5.1 Self-Attention is All You Need

In this section, we analyze how self-attention helps the clustering. As showed in the section 4.3.2, self-attention is designed to construct relevant between two elements in the sequence itself. Eqn. (5) shows the way to compute attention scores between query $\mathbf{Q}_i$ and key $\mathbf{K}_j$. We export the attention map from Clusformer as shown in Figure 7. Overall, the self-attention helps to reduce correlation between hard and positive samples while strengthening the connection within the positive samples.

In Figure 7, the horizontal axis is the query while the vertical axis is the key sequence. Brighter colors will result high attention scores. Firstly, we analyze the attention scores of visual center placed at the first of sequence $\mathbf{Q}_0$ to the rest of others. It is clear that, the attention scores are all lower values. It is noticed that the visual center is always at the fist place and the outputs of Clusformer are ones for all samples as in Section 4.3.1. Therefore, the model does not need to pay attention to this visual element. For others positive samples (orange dot boxes), obviously, the attention scores are high and build up brightness columns at the attention. Even though there are positive samples far away from center (i is high value), the self-attention is able to highlight them clearly. For the hard or noisy samples close to the center (i is the low value), the attention columns (red

Table 1. Performance on face clustering with different number of unlabeled images.

#unlabeled	584K		1.74M		2.89M		4.05M		5.21M		Time
Method / Metrics	F_P	F_B	F_P	F_B	F_P	F_B	F_P	F_B	F_P	F_B	
K-means [21, 31]	79.21	81.23	73.04	75.2	69.83	72.34	67.9	70.57	66.47	69.42	11.5h
HAC [33]	70.63	70.46	54.4	69.53	11.08	68.62	1.4	67.69	0.37	66.96	12.7h
DBSCAN [10]	67.93	67.17	63.41	66.53	52.5	66.26	45.24	44.87	44.94	44.74	**1.9m**
ARO [27]	13.6	17	8.78	12.42	7.3	10.96	6.86	10.5	6.35	10.01	27.5m
CDP [42]	75.02	78.7	70.75	75.82	69.51	74.58	68.62	73.62	68.06	72.92	2.3m
L-GCN [37]	78.68	84.37	75.83	81.61	74.29	80.11	73.7	79.33	72.99	78.6	86.8m
LTC [40]	85.66	85.52	82.41	83.01	80.32	81.1	78.98	79.84	77.87	78.86	62.2m
GCN-V [39]	87.14	85.82	83.49	82.63	81.51	81.05	79.97	79.92	78.77	79.09	4.5m
GCN-VE [39]	87.93	86.09	84.04	82.84	82.1	81.24	80.45	80.09	79.3	79.25	11.5m
Clusformer - Ours	**88.20**	**87.17**	**84.60**	**84.05**	**82.79**	**82.30**	**81.03**	**80.51**	**79.91**	**79.95**	2.2m

Table 2. Performance on Landmark Clustering

Methods	F_P	F_B
K-means [21]	8.52	14.02
HAC [33]	0.2	20.88
DBSCAN [10]	0.97	17.38
Spectral [14]	6.93	18.28
ARO [27]	0.32	10.54
L-GCN [37]	14.08	36.35
GCN - V [39]	16.10	34.86
GCN - VE [39]	10.20	30.23
Clusformer (Ours)	**19.32**	**40.63**

Table 3. Clusformer with Different Encoding Methods

Encoding methods	F_P	F_B
Without Encoding	84.89	83.87
Learnable Encoding	83.00	82.08
Positional Encoding	84.17	82.71
CDE - Ours	**88.20**	**87.17**

Table 4. Average noise ratio remaining in the clusters

Methods	584K	1.75M	2.89M	4.05M	5.21M
GCN-V [39]	0.166	0.199	0.213	0.240	0.269
GCNV-E [39]	0.164	0.193	0.210	0.239	0.267
Clusformer	**0.115**	**0.129**	**0.166**	**0.212**	**0.196**

dot boxes) contain low attention score. Thus, the samples in these positions are likely eliminated from the clusters.

5.5.2 Does Cosine Distance Encoding Really Help?

Our method is experimented with several encoding methods. We select one part, i.e. 584K samples, of MS-Celeb-1M for comparison. The results are reported in the Table 3. There are three scenarios, i.e. without encoding, Learnable Encoding (LE), Positional Encoding (PE) and our Cosine Distance Encoding. For the Learnable Encoding, a learnable $k \times d$ matrix is created during training the model.

It is important to notice that, when we apply LE and PE that are widely used in NLP to this visual problem, they do not work properly. Not using encoding achieves 84.89% of F_P and 83.87% of F_B while LE achieves 83.00% and 82.08%, PE gets 84.17% and 82.71% for F_P and F_B, respectively. The reason is that the input sequence has been well defined with a visual grammar, so it is not necessary to use NLP encoding methods. Instead, the proposed CDE is more appropriate for visual clustering problem. It significantly boost the performance improvement compared against the other NLP encoding methods by achieving 88.20% of F_P and 87.17% of F_B. It strongly shows that our CDE is reasonable and effective.

5.5.3 Robustness of Clusformer to Noise.

To further illustrate the noise robustness of the method, we split MS1M into multiple parts ranging from 584K to 5.21M samples and measure the ratio of hard/noise sample remaining inside the cluster extracted by Clusformer (Table 4). These results re-emphasize the advantage of Clusformer in removing more noisy/hard samples than both GCNV and GCNVE with margins of 4%-7%.

6. Conclusions

We have presented a novel method, namely Clusformer, for visual clustering problems, i.e. large-scale face and landmark clustering, thanks to its self-attention mechanisms. Besides, a new visual grammar has presented to construct visual sequence from a given cluster. An innovative encoding method called Cosine Distance Encoding is also introduced to improve the performance of Clusformer. Our framework achieves the state-of-the-art results in both face and visual landmark clustering and recognition problems.

References

[1] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2):49–60, 1999. 2

[2] Bingyi Cao, Andre Araujo, and Jack Sim. Unifying deep local and global features for efficient image search. *arXiv preprint arXiv:2001.05027*, 2020. 2

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020. 5

[4] Junjie Chen, William K Cheung, and Anran Wang. Aphash: Anchor-based probability hashing for image retrieval. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1673–1677. IEEE, 2018. 2

[5] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020. 3

[6] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019. 3

[7] Jiankang Deng, Jia Guo, Tongliang Liu, Mingming Gong, and Stefanos Zafeiriou. Sub-center arcface: Boosting face recognition by large-scale noisy web faces. In *Proceedings of the IEEE Conference on European Conference on Computer Vision*, 2020. 1

[8] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 6

[9] 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. 3

[10] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996. 6, 8

[11] Jules. Harvey, Adam. LaPlace. Megapixels.cc: Origins, ethics, and privacy implications of publicly available face recognition image datasets, 2019. 5

[12] 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. 6

[13] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020. 3

[14] Jeffrey Ho, Ming-Husang Yang, Jongwoo Lim, Kuang-Chih Lee, and David Kriegman. Clustering appearances of objects under varying illumination conditions. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 1, pages I–I. IEEE, 2003. 1, 8

[15] Yuge Huang, Yuhan Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, and Feiyue Huang Jilin Li. Curricularface: Adaptive curriculum learning loss for deep face recognition. pages 1–8, 2020. 1

[16] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019. 6

[17] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4873–4882, 2016. 6

[18] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 1, 2

[19] Wei-An Lin, Jun-Cheng Chen, and Rama Chellappa. A proximity-aware hierarchical clustering of faces. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 294–301. IEEE, 2017. 2

[20] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 3

[21] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982. 2, 6, 8

[22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[23] Khoa Luu. A computer approach for face aging problems. pages 405–409, 05 2010. 1

[24] Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. Camembert: a tasty french language model. *arXiv preprint arXiv:1911.03894*, 2019. 3

[25] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE international conference on computer vision*, pages 3456–3465, 2017. 2

[26] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828, 2018. 2

[27] Charles Otto, Dayong Wang, and Anil K Jain. Clustering millions of faces by identity. *IEEE transactions on pattern analysis and machine intelligence*, 40(2):289–303, 2017. 1, 2, 6, 8

[28] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018. 3

[29] Jerome Revaud, Jon Almazán, Rafael S Rezende, and Cesar Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5107–5116, 2019. 2

[30] Fabio Roli and Gian Luca Marcialis. Semi-supervised pca-based face recognition using self-training. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 560–568. Springer, 2006. 2

[31] David Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178, 2010. 6, 8

[32] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000. 2

[33] Robin Sibson. Slink: an optimally efficient algorithm for the single-link cluster method. *The computer journal*, 16(1):30–34, 1973. 6, 8

[34] Marvin Teichmann, Andre Araujo, Menglong Zhu, and Jack Sim. Detect-to-retrieve: Efficient regional aggregation for image search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5109–5118, 2019. 2

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 3, 5

[36] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018. 1

[37] Zhongdao Wang, Liang Zheng, Yali Li, and Shengjin Wang. Linkage based face clustering via graph convolution network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1117–1125, 2019. 2, 7, 8

[38] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2575–2584, 2020. 1, 6

[39] Lei Yang, Dapeng Chen, Xiaohang Zhan, Rui Zhao, Chen Change Loy, and Dahua Lin. Learning to cluster faces via confidence and connectivity estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2, 3, 4, 6, 7, 8

[40] Lei Yang, Xiaohang Zhan, Dapeng Chen, Junjie Yan, Chen Change Loy, and Dahua Lin. Learning to cluster faces on an affinity graph. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 3, 4, 6, 7, 8

[41] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763, 2019. 3

[42] Xiaohang Zhan, Ziwei Liu, Junjie Yan, Dahua Lin, and Chen Change Loy. Consensus-driven propagation in massive unlabeled data for face recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 568–583, 2018. 2, 6, 7, 8

[43] Xuran Zhao, Nicholas Evans, and Jean-Luc Dugelay. Semi-supervised face recognition with lda self-training. In *2011 18th IEEE International Conference on Image Processing*, pages 3041–3044. IEEE, 2011. 2