Neural Clustering based Visual Representation Learning

Supplementary Material

For a better understanding of the main paper, we provide additional details in this supplementary material, which is organized as follows:

- §A provides the pseudo code of FEC.
- §B introduces more experimental details.
- §C offers more results and discussions about the modeled representatives.
- §D discusses our limitations, societal impact, and directions of future work.

A. Pseudo Code

To facilitate a comprehensive understanding of FEC, we provide pseudo code for our feature encoding and feature pooling in Algorithm S1.

B. More Experimental Detail

Image Classification. In this task, several widely-used data augmentations are adopted to better train the model, including random horizontal flipping, random pixel erase [1], MixUp [2], CutMix [3], and label smoothing [4]. We employ an AdamW [5] optimizer using a cosine decay learning rate scheduler and 5 epochs of warm-up. The momentum and weight decay are set to 0.9 and 0.05, respectively. A batch size of 1024 and an initial learning rate of 0.001 are used. We also use exponential moving average [6] to enhance the training. Throughput (image/s), or FPS, is measured using the same script [7, 8] on a single V100 GPU using a batch size of 256. The reported values are averaged by 100 iterations after 20 warm iterations. We use the same codebase and tricks (*e.g.*, multi-head computing) as in [9]. In addition, we use almost the same hyperparameters and architectures as in [9] for fair comparison.

Downstream Tasks. During training, backbones are initialized with weights pre-trained on ImageNet [10], while the other parts are initialized randomly.

C. Modeled Representative

In the "Study of *Ad-hoc* Interpretability" section of the main paper, it is highlighted that FEC's final cluster assignments display consistent semantic representations. These representations frequently correlate with distinct objects or their components and demonstrate a close alignment with human perception. Here we visualize more results of cluster assignments in Fig. S1 to clarify the FEC's principles. Similar conclusions can be drawn from Fig. S1, which confirms again the *ad-hoc* interpretability and effectiveness of FEC.

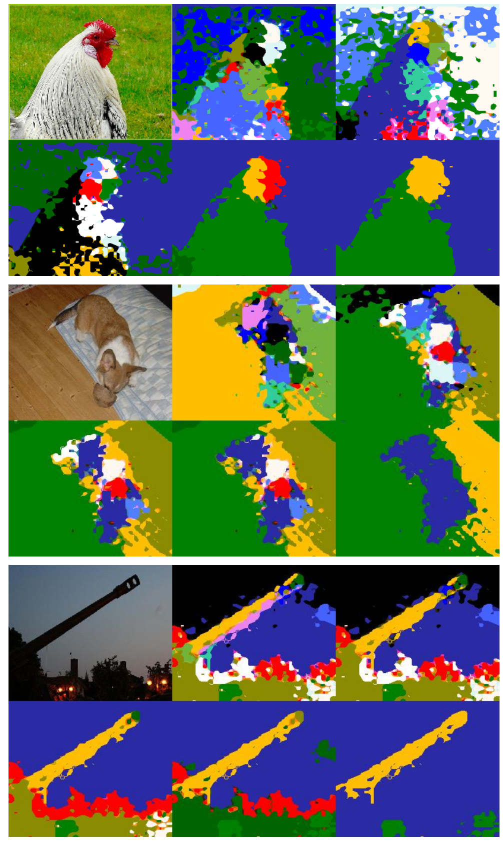

Figure S1. Inspection of the modeled representatives (§C) on ImageNet-1K [10] `val`.

D. Discussion

Limitation Analysis. One limitation of our approach is the adoption of a straightforward clustering mechanism, primarily aimed at ensuring computational efficiency. While this design choice contributes to faster processing times, it may inadvertently lead to sub-optimal performance in certain scenarios. Additionally, akin to many parametric clustering algorithms [11–13], our method requires the manual definition of the number of clusters to keep the same resolution with previous works [7, 14–16]. This aspect introduces a degree of subjectivity and potential bias, as the predetermined cluster count may not align perfectly with the intrinsic structure of specific images, particularly in dealing with datasets where the optimal number of clusters is not known a priori or varies significantly.

Societal Impact. This work provides a clustering perspective for transparent, *ad-hoc* interpretable feature extraction,

Algorithm S1 Pseudo code of FEC in a PyTorch-like style.

```
# feat_i: input feature (N x C), where N = W x H

# C: number of channels
# N: resolution of input feature
# O: number of cluster centers. In pooling, O = N/4. In encoding, O is a hyperparameter (O < N).
# M: similarity matrix (Eq.5)
# A: cluster assigment matrix
# R: representatives
# sig: sigmoid function
# alpha and beta: learnable parameters

def model_representatives(feat_i)
   # center initialization (Eq.4)
   feat_k = conv_k(feat_i) # (N x C')
   feat_v = conv_v(feat_i) # (N x C')
   feat_c_k = ada_pool(feat_k) # (O x C')
   feat_c_v = ada_pool(feat_v) # (O x C')

   # compute similarities and cluster assigments (Eq.5)
   M = cosine_sim(feat_k, feat_c_k) # (N x O)
   A = torch.argmax(M, dim=1) # (N x O)

   # aggragate the feature of representatives (Eq.6)
   R = aggragate_feature(feat_v, feat_c_v, A) # (O x C')

   return R, M

def pooling(feat_i)
   R, _ = model_representatives(feat_i)
   res_conn = ResConn(feat_i) # residual connection (Eq.9)

   return R + res_conn

def encoding(feat_i)
   R, M = model_representatives(feat_i)

   # feature dispatching (Eq.7)
   refined_M = sig(alpha * M + beta).permute(1,0) # (O x N)
   feat_d = ( R.unsqueeze(dim=1) * refined_M.unsqueeze(dim=-1) ).sum(dim=0) # (N x C')
   feat_d = MLP(feat_d) # (N x C)
   out = feat_i + feat_d # residual connection

   return out
```

and accordingly introduces a novel visual backbone which reformulates the entire process of feature extraction as representative selection. On positive side, the approach advances network interpretability and is valuable in safety-sensitive applications, *e.g.*, medical image analysis [17], face recognition [18, 19], and autonomous driving [20, 21]. For potential negative social impact, the erroneous recognition may cause inaccurate decision or planning of systems based on the results. In addition, the potential bias inherent in the training data may be exploited for malicious purposes.

Future Work. This work also comes with new challenges, certainly worth further exploration:

- **Incorporating Advanced Clustering Algorithms**. In future developments, we aim to augment the FEC framework by incorporating advanced clustering algorithms. Our current model prioritizes computational efficiency with a straightforward clustering mechanism, but we recognize opportunities for enhancing performance and accuracy. Upcoming versions will investigate sophisticated algorithms adept at managing complex data structures and distributions, potentially increasing the granularity and precision of feature extraction for more refined and accurate visual representations. An intriguing avenue is transitioning from parametric clustering, which presupposes a fixed number of clusters, to nonparametric clustering, where the number of clusters is undetermined. There are numerous techniques for nonparametric clustering, including Bayesian nonparametric (BNP) mixture models (exemplified by the Dirichlet Process Mixture (DPM) model [22, 23]), DPM sampler [24–27], variational DPM inference [28–32], density-based approach [33], nearest-neighbor graph [34], supervised approach [35, 36], dynamic network architecture [37]. We have explored a very recent work, *i.e.*, DeepDPM [37]. However, after running their code, we find that DeepDPM is notably complex and require substantial computational time. Moving forward, our focus is on identifying better trade-offs between complexity, computational efficiency, and performance.
- **Combination with Set-prediction Architectures**. The recent emergence of set-prediction architectures, such

as DETR [38], presents a significant opportunity to utilize the representatives modeled by FEC more effectively. Unlike traditional methods that rely on handcrafted components like non-maximum suppression for post-processing and pre-defined anchors for label assignments, these approaches simplify the pipeline by allowing for end-to-end training and inference. This reduces the need for many of the specialized components typically used in object detection systems and provides an ideal framework for utilizing the representatives extracted by FEC. For example, the modeled representatives can be applied as a metric for distance measurement, aiding in the stabilization of bipartite matching. This integration effectively infuses the concept of "instances" (or representatives) into the feature extraction process, which stands as the primary motivation behind this work.

References

[1] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, pages 13001–13008, 2020. 1

[2] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 1

[3] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6023–6032, 2019. 1

[4] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016. 1

[5] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[6] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992. 1

[7] 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 *ICCV*, 2021. 1

[8] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357, 2021. 1

[9] Xu Ma, Yuqian Zhou, Huan Wang, Can Qin, Bin Sun, Chang Liu, and Yun Fu. Image as set of points. In *ICLR*, 2023. 1

[10] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 1

[11] Tom M Mitchell. Machine learning and data mining. *Communications of the ACM*, 42(11):30–36, 1999. 1

[12] Rui Xu and Donald Wunsch. Survey of clustering algorithms ieee transactions on neural networks, vol. 16 (3), 2005.

[13] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *ECCV*, 2020. 1

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[15] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. Resmlp: Feedforward networks for image classification with data-efficient training. *IEEE TPAMI*, 45(4):5314–5321, 2022.

[16] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *CVPR*, pages 10819–10829, 2022. 1

[17] James S Duncan and Nicholas Ayache. Medical image analysis: Progress over two decades and the challenges ahead. *TPAMI*, 22(1):85–106, 2000. 2

[18] Xiao Yang, Fangyun Wei, Hongyang Zhang, and Jun Zhu. Design and interpretation of universal adversarial patches in face detection. In *ECCV*, pages 174–191, 2020. 2

[19] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. Efficient decision-based blackbox adversarial attacks on face recognition. In *CVPR*, pages 7714–7722, 2019. 2

[20] Yaqin Wang, Dongfang Liu, Hyewon Jeon, Zhiwei Chu, and Eric T Matson. End-to-end learning approach for autonomous driving: A convolutional neural network model. In *ICAART*, 2019. 2

[21] Dongfang Liu, Yiming Cui, Xiaolei Guo, Wei Ding, Baijian Yang, and Yingjie Chen. Visual localization for autonomous driving: Mapping the accurate location in the city maze. In *ICPR*, 2021. 2

[22] Charles E Antoniak. Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The annals of statistics*, pages 1152–1174, 1974. 2

[23] Thomas S Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973. 2

[24] Jason Chang and John W Fisher III. Parallel sampling of dp mixture models using sub-cluster splits. In *NeurIPS*, 2013. 2

[25] Gang Chen. Deep learning with nonparametric clustering. *arXiv preprint arXiv:1501.03084*, 2015.

[26] Or Dinari, Angel Yu, Oren Freifeld, and John Fisher. Distributed mcmc inference in dirichlet process mixture models using julia. In *2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pages 518–525, 2019.

[27] Zeya Wang, Yang Ni, Baoyu Jing, Deqing Wang, Hao Zhang, and Eric Xing. Dnb: A joint learning framework for deep bayesian nonparametric clustering. *TNNLS*, 33(12):7610–7620, 2021. 2

[28] David M Blei and Michael I Jordan. Variational inference for dirichlet process mixtures. *Bayesian analysis*, 2006. 2

[29] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *JMLR*, 2013.

[30] Michael C Hughes and Erik Sudderth. Memoized online variational inference for dirichlet process mixture models. In *NeurIPS*, 2013.

[31] Viet Huynh, Dinh Phung, and Svetha Venkatesh. Streaming variational inference for dirichlet process mixtures. In *ACML*, pages 237–252, 2016.

[32] Kenichi Kurihara, Max Welling, and Nikos Vlassis. Accelerated variational dirichlet process mixtures. In *NeurIPS*, 2006. 2

[33] 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*, pages 226–231, 1996. 2

[34] Sohil Atul Shah and Vladlen Koltun. Deep continuous clustering. *arXiv preprint arXiv:1803.01449*, 2018. 2

[35] Makarand Tapaswi, Marc T Law, and Sanja Fidler. Video face clustering with unknown number of clusters. In *ICCV*, pages 5027–5036, 2019. 2

[36] Ari Pakman, Yueqi Wang, Catalin Mitelut, JinHyung Lee, and Liam Paninski. Neural clustering processes. In *ICML*, pages 7455–7465, 2020. 2

[37] Meitar Ronen, Shahaf E Finder, and Oren Freifeld. Deepdpm: Deep clustering with an unknown number of clusters. In *CVPR*, 2022. 2

[38] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 3