Supplementary Material: Anti-Degradation Lifelong Multi-View Clustering

Xingfeng Li[1], Hao Pan[1], Honglin Yuan[1], Yuan Sun[2]*, Xujian Zhao[1], Jiaqi Lin[3], Zhenwen Ren[1]†
[1] Southwest University of Science and Technology, Mianyang, China 621010
[2] Sichuan University, Chengdu, China 610044
[3] Agency for Science, Technology and Research, Singapore 138632
lixingfeng@njust.edu.cn, pon915@163.com, hl_yuan0822@163.com, sunyuan_work@163.com, jasonzhaoxj@swust.edu.cn, linjq56@mail2.sysu.edu.cn, rzw@njust.edu.cn

1. Summary

In the supplementary material, we provide additional implementation details and experimental results. Section 2, 6 present supplementary information, and Section 3, 4, 5 include additional experimental results.

2. Experiment Settings

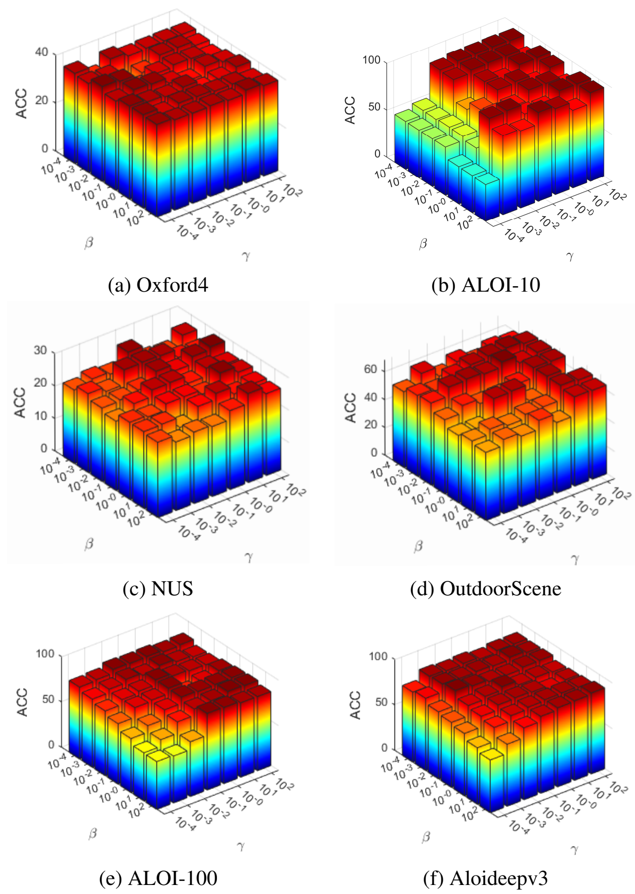
In the ALMC algorithm, we introduce two tunable hyperparameters, *i.e.*, β and γ. For the ALOI-10 and NUS datasets, we set $\beta = 1, \gamma = 1$ and $\beta = 1, \gamma = 0.1$, respectively. For the Outdoorscene dataset, β and γ are set to 0.1 and 0.01. For all remaining datasets, both hyperparameters are fixed to 0.1. To ensure a comprehensive evaluation of clustering performance, all methods are tested using five different random seeds and evaluation runs.

3. Parameter Sensitive Analysis

ALMC incorporates two adjustable hyperparameters, β and γ, which respectively regulate the reconstruction loss and the knowledge-guided learning loss. To examine the impact of these hyperparameters, we explore values spanning from 10^2 down to 10^{-4}. As illustrated in Fig. 1, setting either parameter too high or too low negatively affects clustering performance, emphasizing the necessity of appropriately balancing all three losses in the model. Based on our empirical findings, we suggest selecting values within the range of 10^1 to 10^{-1} for both parameters.

4. View Sensitivity Analysis

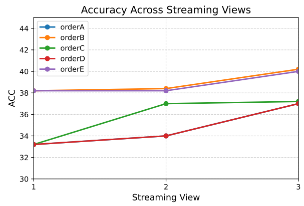
This section provides additional results illustrating the impact of different streaming-view orders on model performance, as shown in Fig. 2. When processing views arriving in different orders, the initial performance gaps are noticeable. This is primarily because the early-arriving views exhibit substantial differences in their underlying data distributions. However, as more views are accumulated, these gaps gradually diminish. After each view is processed, its learned prototypes are projected into the null space and incorporated into the knowledge space, where the knowledge-guided learning module progressively strengthens the clus-



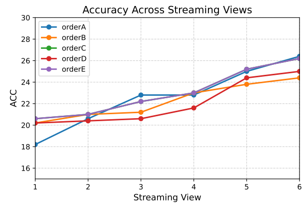
(a) Oxford4 (b) ALOI-10 (c) NUS (d) OutdoorScene (e) ALOI-100 (f) Aloideepv3

Figure 1. Parameter sensitivity analysis on six datasets.

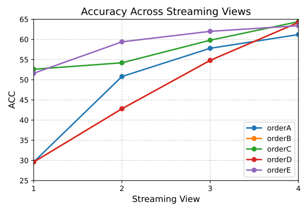
*Corresponding author.
†Corresponding author.



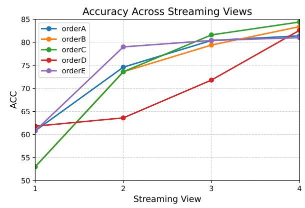
(a) Oxford4



(b) NUS

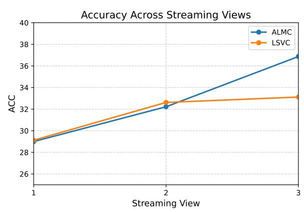


(c) OutdoorScene

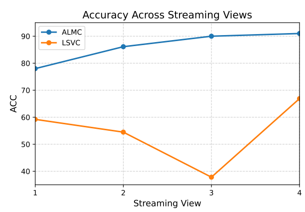


(d) ALOI-100

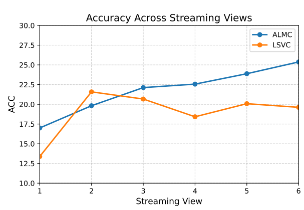
Figure 2. View sensitivity analysis on four datasets.



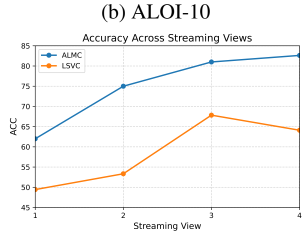
(a) Oxford4



(b) ALOI-10



(c) NUS



(d) ALOI-100

Figure 3. Clustering performance comparison on four datasets.

tering structure. Consequently, the final performance converges to a stable and consistent level across different ordering scenarios.

5. View Streaming Analysis

This section presents additional experimental results on the continuous accumulation of views, as shown in Fig. 3. On the Oxford4 dataset, both ALMC and LSVC exhibit improved clustering performance as more views are added. However, LSVC shows only marginal performance gains in the last two views, indicating that it fails to fully leverage the informative knowledge contained in earlier views. In contrast, ALMC not only achieves steady improvements but also demonstrates consistently large performance gains across different views. For the other three datasets, when newly collected views overlap with the knowledge already represented in the existing feature space, the performance of LSVC drops noticeably. In comparison, ALMC continues to improve steadily, benefiting from our null-space projection strategy. This mechanism ensures that newly uploaded knowledge is updated along novel directions without interfering with previously stored knowledge, while also preserving the intrinsic clustering structure of the learned representations. As a result, ALMC achieves superior and more stable clustering performance.

6. ALMC Training Algorithm

In this section, we present the main algorithmic process of ALMC, as shown in Algorithm 1.

Algorithm 1 The algorithm of ALMC

Require: New streaming view data X^v with size N; Batch size S; Training epoch E; The number of prototype knowledge is K.
Ensure: Parameters: β and $\gamma 0$.

```
if v = 1 then
    while epoch < E do
        for i = 1 to N/S do
            Training the network.
            Aggregate prototype knowledge P^v and features H^v.
            Leverages prototype knowledge P^v to guide the feature H^v distribution.
        end for
        Update prototype knowledge P^v into knowledge space B.
    end while
    Perform K-means algorithm on H^v.
end if
if v > 1 then
    while epoch < E do
        for i = 1 to N/S do
            Training the network.
            Aggregate prototype knowledge P^v and features H^v.
            Align the current view prototype knowledge P^v with the knowledge base B distribution.
            Leverages prototype knowledge P^v to guide the feature H^v distribution.
        end for
        After the null-space projection, the prototypes P^v are uploaded to the knowledge space B.
    end while
    Project H^v onto the null space of H^{v-1}, and then concatenate the projected representation with H^{v-1}.
    Perform K-means algorithm.
end if
```