# TALON: Test-time Adaptive Learning for On-the-Fly Category Discovery

Supplementary Material

## A. Implementation Details

### A.1. Datasets Details and Evaluation Metric Details

**Dataset Details.** To comprehensively evaluate our framework across different levels of semantic complexity, we adopt a diverse set of benchmark datasets that cover both coarse-grained and fine-grained recognition scenarios.As shown in Tab. 5, the coarse-grained datasets used in our experiments include CIFAR-10 [25], CIFAR-100 [25], and ImageNet-100 [45].These datasets exhibit large inter-class variation and broad semantic categories, making them suitable for evaluating category discovery under generic object recognition settings. In contrast, the fine-grained datasets summarized in Tab. 6. CUB-200-2011 [49], Stanford Cars [24],Oxford Pets [39], and Food-101 [1] contain subtle visual distinctions among classes and therefore provide a more challenging testing ground for fine-grained category discovery. By jointly employing datasets of different granularity, our evaluation ensures that the proposed method is rigorously tested under diverse and realistic visual recognition conditions.Specifically, 50% of the samples from the seen categories are used to form the labeled training set $\mathcal{D}_S$, while the remainder forms the unlabeled set $\mathcal{D}_Q$ for on-the-fly testing.

Table 5. Statistics of coarse-grained datasets.

| Dataset | CIFAR10 | CIFAR100 | ImageNet-100 |
|---|---|---|---|
| $\lvert Y_Q\rvert$ | 5 | 80 | 80 |
| $\lvert Y_S\rvert$ | 10 | 100 | 100 |
| $\lvert D_S\rvert$ | 12.5K | 20.0K | 31.9K |
| $\lvert D_Q\rvert$ | 37.5K | 30.0K | 95.3K |

Table 6. Statistics of fine-grained datasets.

| Dataset | CUB-200-2011 | Stanford Cars | Oxford Pets | Food101 |
|---|---|---|---|---|
| $\lvert Y_Q\rvert$ | 100 | 98 | 19 | 51 |
| $\lvert Y_S\rvert$ | 200 | 196 | 38 | 101 |
| $\lvert D_S\rvert$ | 1.5K | 2.0K | 0.9K | 19.1K |
| $\lvert D_Q\rvert$ | 4.5K | 6.1K | 2.7K | 56.6K |

**Evaluation Metric Details.** Following [11], we adopt two protocols for evaluation termed *Greedy-Hungarian* and *Strict-Hungarian* for comprehensive comparisons, where their difference is clearly illustrated in [48]. During testing, samples sharing the same category descriptor form a cluster, and only the top-$|\mathcal{Y}_Q|$ clusters by size are retained, with the rest treated as misclassified. For *Greedy-Hungarian*, accuracy is computed separately on the "Old" and "New" subsets, providing independent evaluations of known and novel classes. In contrast, *Strict-Hungarian* calculates accuracy over the entire query set, avoiding repeated cluster assignments between subsets. The overall accuracy is obtained via the Hungarian matching:

$$ACC = \max_{p \in \mathcal{P}(\mathcal{Y}_Q)} \frac{1}{|\mathcal{D}_Q|} \sum_{i=1}^{|\mathcal{D}_Q|} \mathbb{1}[y_i = p(\hat{y}_i)], \tag{20}$$

where $y_i$ is the ground truth label, $\hat{y}_i$ is the predicted label decided by cluster indices, and $\mathcal{P}(\mathcal{Y}_Q)$ is the set of all permutations of ground truth lables.

### A.2. Training Details

We adopt two vision backbones in our experiments, namely CLIP ViT-B/16 and DINOv2-ViT-Base, and build our framework on top of their pre-trained weights. For CLIP, we use the publicly released ViT-B/16 model, who outputs 512-dimensional image embeddings. For DINOv2, we use the official DINOv2-ViT-Base checkpoint, its output features are further projected into the 768-dimensional visual space.

We train the model using the AdamW optimizer with a weight decay of 0.05 and a cosine learning rate scheduler with a minimum learning rate of $1e{-}5$. Following our implementation, we employ parameter-specific learning rates: the prototype layer is optimized with a learning rate of $1e{-}3$, the last visual layer of the backbone is updated with a smaller learning rate of $1e{-}4$, and the remaining parameters follow the base learning rate governed by the scheduler. The batch size is uniformly set to 128 for all datasets, and we train for 100 epochs. All experiments are conducted on NVIDIA RTX 3090 GPUs with 24GB memory, and we fix the random seed to 1028 to improve reproducibility.

### A.3. Compared Methods Details

**Ranking Statistics (RankStat).** [16] AutoNovel adopts Ranking Statistics to characterize sample relationships, where the top-3 indices of feature embeddings are taken as category indicators. This formulation fits well within the On-the-Fly Category Discovery (OCD) paradigm and presents a strong benchmark for evaluating hash-based descriptors. For fairness, we reimplement Ranking Statistics using the same backbone (DINO-ViT-B-16) and preserve only the fully supervised training stage, since OCD does not allow any external data. The embedding dimension is fixed to 32, yielding a prediction space of $C_{32}^3 = 4,906$, which is comparable to our approach and SMILE, where a hash length of $L = 12$ produces $2^{12} = 4,096$ possible codes.

**Winner-take-all (WTA).** [21] To mitigate the bias of Ranking Statistics toward overly prominent features, the

Winner-take-all (WTA) hashing strategy was proposed. Instead of relying on a global ranking of feature activations, WTA constructs descriptors by identifying the maximum index within each of several partitioned feature groups. Using a 48-dimensional embedding split into three groups, WTA generates $16^3 = 4096$ distinct codes, ensuring direct comparability with the other hashing-based baselines.

**Sequential Leader Clustering (SLC).** [17] For SLC, we adopt the same backbone and apply conventional supervised learning on the support set. During on-the-fly evaluation, SLC assigns labels using features extracted from the query samples. Hyperparameters are tuned on the CUB dataset and subsequently applied unchanged to all other datasets to maintain consistency and fairness across comparisons.

**MLDG.** [26] Different from the standard NCD setting, which leverages both support and query sets to learn discriminative features, OCD is closer to a domain-generalization problem: the model is trained on seen categories and must generalize to unseen ones at inference time. Therefore, we include MLDG [26], a model-agnostic domain generalization algorithm, as a strong competitor. During training, samples from different classes are partitioned into meta-train and meta-test domains at each iteration to encourage generalizable representations.

For **SMILE**[11], **PHE**[61] and **DiffGRE**[33], the best configuration of their original paper reports is adopted

## B. Additional Experiment and Analysis

### B.1. Hyperparameter analysis.

We further analyze the sensitivity of our method to two key hyper-parameters: the logit scale $s$ and the smoothing constant $\kappa$. The $s$ controls the magnitude of the logits in MLC, thereby affecting the margin strength and confidence of the classifier, while $\kappa$ appears in the Semantic-aware prototype update and balances how fast the prototypes are allowed to adapt to incoming test samples. To assess the robustness of our method, we conduct a series of experiments by varying $s$ in $\{10, 15, 20, 25, 30, 35, 40, 45, 50\}$ and $\kappa$ in $\{4, 5, 6, 7, 8, 9, 10, 11, 12\}$, and report the corresponding results in Fig. 5.

### B.2. The influence of different backbone networks

To comprehensively evaluate the adaptability and robustness of our approach, we compare it against two representative state-of-the-art methods—SMILE and PHE—under two widely adopted visual backbones, CLIP and DINO,as shown in Tab. 7 Specifically, CLIP is a large-scale vision–language contrastive model known for its strong generalization ability, especially in open-world scenarios, while DINO is a self-supervised representation learning method whose distilled features are highly structured and discriminative, offering superior intra-class compactness and inter-class separability.

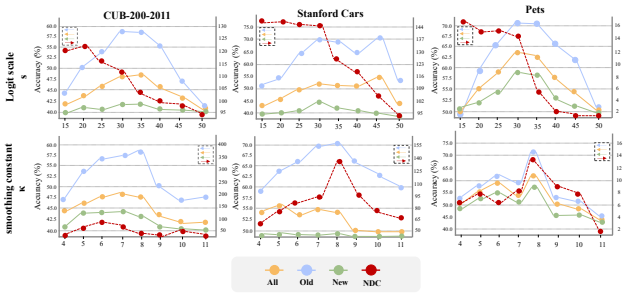


Figure 5. **Hyperparameter analysis on CUB-200-2011, Stanford Cars, and Oxford Pets datasets.** Each column corresponds to one dataset, showing the effects of logit scale $s$ and the smoothing constant $\kappa$ on accuracy (*All*, *Old*, *New*) and the number of newly discovered categories (NDC).

Across both backbones, our method consistently outperforms SMILE and PHE on all benchmarks. The improvements are observed not only in the overall accuracy (All) but also in the recognition of Old and particularly New categories, where our gains are most prominent. These results demonstrate that our proposed matching strategy remains effective across different feature spaces, showing stronger generalization and robustness compared to existing approaches.

In summary, the experiments with multiple backbones further validate the stability and superiority of our method under diverse representation settings.

### B.3. Computational Complexity Analysis.

In our framework, the OCD task can be decomposed into three stages: a training stage, a prototype computation stage, and a test stage. Using CLIP as the visual backbone, we measure the running time of each stage and repeat the experiment five times independently. The averaged results and standard deviations are reported in Tab. 8.All experiments are conducted on a single NVIDIA RTX 3090 GPU with 24 GB memory.

Table 8. Three-stage time consumption proportion display table

| Dataset | Train (ms) | Prototype computation (ms) | Test (ms) |
|---|---|---|---|
| CIFAR-10 | 746.506 ± 1.445 | 667.455 ± 41.925 | 1496.276 ± 37.384 |
| CIFAR-100 | 747.667 ± 3.647 | 734.105 ± 55.240 | 1504.007 ± 65.634 |
| ImageNet-100 | 746.663 ± 1.120 | 666.157 ± 3.393 | 2324.649 ± 535.563 |
| CUB-200-2011 | 748.577 ± 1.020 | 642.528 ± 1.003 | 1753.667 ± 44.756 |
| Stanford Cars | 750.333 ± 3.086 | 653.369 ± 1.991 | 1736.116 ± 64.858 |
| Oxford Pets | 749.094 ± 3.262 | 641.062 ± 0.394 | 1472.476 ± 72.950 |
| Food101 | 750.889 ± 0.596 | 647.426 ± 2.539 | 1724.313 ± 17.371 |

As shown in the Fig. 6, the time proportion during the test is the longest among the three stages. This is due to the introduction of the TTA algorithm. However, under the premise of improving such high accuracy, we believe it is worthwhile.

Table 7. **Comparison with State-of-the-Art Methods Using Different Backbones**

| | Method | Backbone | CIFAR10 (%) | | | CIFAR100 (%) | | | ImageNet-100 (%) | | | CUB-200-2011 (%) | | | Stanford Cars (%) | | | Oxford Pets (%) | | | Food101 (%) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | All | Old | New | All | Old | New | All | Old | New | All | Old | New | All | Old | New | All | Old | New | All | Old | New |
| *Greedy–Hungarian* | SMILE [11] | DINO | 78.2 | **99.3** | 67.6 | 61.3 | 70.7 | 42.5 | 39.9 | 87.1 | 16.2 | 41.1 | 67.6 | 27.8 | 33.4 | 58.4 | 21.3 | 54.1 | 66.1 | 47.8 | 34.4 | 64.0 | 19.4 |
| | SMILE [11] | CLIP | 82.4 | 97.4 | 74.9 | 56.4 | 64.6 | 40.0 | 47.5 | 71.0 | 35.7 | 43.7 | 69.7 | 30.8 | 36.7 | 57.2 | 26.8 | 58.2 | 77.5 | 48.1 | 40.5 | 70.4 | 25.2 |
| | PHE [61] | DINO | 83.0 | 98.0 | 75.5 | 64.8 | 78.8 | 36.9 | 53.1 | 83.5 | 38.1 | 46.9 | 76.0 | 32.4 | 46.3 | 78.3 | 30.8 | 63.3 | 91.3 | 48.6 | 50.0 | **89.3** | 30.0 |
| | PHE [61] | CLIP | 79.3 | 97.0 | 70.4 | 66.1 | 80.3 | 37.5 | 52.9 | 87.8 | 35.5 | 44.2 | 70.3 | 31.1 | 46.4 | 78.1 | 31.1 | 64.1 | 86.2 | 52.4 | 47.8 | 88.4 | 27.0 |
| | **Ours** | DINO | **86.2** | 95.4 | **79.3** | **72.5** | **85.2** | **47.0** | **84.1** | 94.3 | **63.4** | 52.6 | 83.3 | 37.2 | 42.9 | 78.1 | 25.9 | **81.0** | 92.4 | **75.1** | 44.5 | 80.4 | 26.2 |
| | **Ours** | CLIP | 84.7 | 96.0 | 76.3 | 69.9 | 82.8 | 44.2 | 83.6 | **96.1** | 58.2 | **58.9** | **87.3** | **44.7** | **60.4** | **90.6** | **45.8** | 74.9 | **94.6** | 64.6 | **61.2** | 88.3 | **47.3** |
| *Strict–Hungarian* | SMILE [11] | DINO | 49.9 | 39.9 | 54.9 | 51.6 | 61.6 | 31.7 | 33.8 | 74.2 | 13.5 | 32.2 | 50.9 | 22.9 | 26.2 | 46.6 | 16.3 | 42.9 | 38.7 | 45.1 | 24.2 | 54.3 | 8.8 |
| | SMILE [11] | CLIP | 51.9 | 19.7 | 68.0 | 46.7 | 55.3 | 29.5 | 35.7 | 41.4 | 32.8 | 34.7 | 55.2 | 24.5 | 32.4 | 46.2 | 25.7 | 40.3 | 37.4 | 41.8 | 33.3 | 56.3 | 21.5 |
| | PHE [61] | DINO | 53.1 | 19.3 | 70.0 | 56.0 | 70.1 | 27.8 | 39.2 | 49.3 | 34.1 | 36.4 | 55.8 | 27.0 | 31.3 | 61.9 | 16.8 | 48.3 | 53.8 | 45.4 | 29.1 | 64.7 | 11.1 |
| | PHE [61] | CLIP | 52.4 | 18.3 | 69.5 | 56.8 | 71.9 | 26.5 | 39.2 | 60.7 | 28.4 | 35.1 | 54.5 | 25.4 | 36.2 | 54.2 | 27.4 | 52.0 | 52.3 | 51.9 | 33.5 | 58.6 | 20.6 |
| | **Ours** | DINO | **65.0** | **46.1** | **79.3** | **64.7** | **77.4** | **39.3** | **82.6** | 92.0 | **63.4** | 43.6 | **63.5** | 33.6 | 37.0 | 68.1 | 22.0 | **69.2** | 58.5 | **74.8** | 30.3 | 60.5 | 15.0 |
| | **Ours** | CLIP | 56.9 | 31.1 | 76.3 | 61.6 | 75.0 | 34.9 | 80.9 | **93.6** | 54.9 | **45.5** | 60.7 | **37.8** | **53.5** | **74.2** | **43.6** | 64.0 | **65.4** | 63.3 | **50.3** | **66.2** | **42.2** |

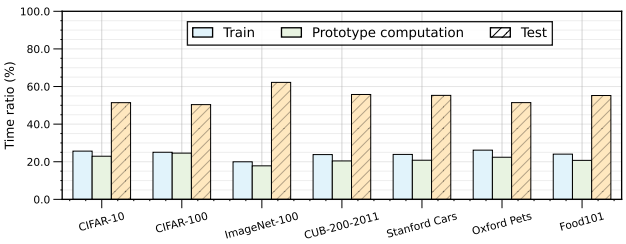


Figure 6. **Display of the proportion of time consumed in the three stages.**

Table 9. Comparison of training times (in second)

| Method | CUB | SCars | Food | Pets |
|---|---|---|---|---|
| SMILE | 4204.9 | 5846.7 | 27000.7 | 2646.5 |
| PHE | 3300.0 | 5313.6 | 22765.7 | 2287.9 |
| Ours | **1452.3** | **1823.2** | **12157.5** | **1074.5** |

## B.4. Training Efficiency Analysis

We provide a comparison of training times between our PHE and the state-of-the-art method, SMILE, with results shown in Table 9. To ensure fairness, all experiments are conducted on an NVIDIA RTX 3090 GPU, with both algorithms trained over 100 epochs using mixed precision. The dataloader parameters are kept consistent across tests, with a batch size of 128.

According to the results in Table 9, our method achieves consistently lower training time across all four datasets compared with both SMILE and PHE. Relative to SMILE, our approach reduces training time by 2752.6 s on CUB, 4023.5 s on SCars, 14843.2 s on Food, and 1572.0 s on Pets. Compared with PHE, our method is also substantially more efficient, yielding reductions of 1847.7 s, 3490.4 s, 10608.2 s, and 1213.4 s on the four datasets, respectively. These results demonstrate that our method provides the fastest training among all compared approaches. This improvement largely stems from our more efficient feature learning strategy, whereas SMILE incurs heavy computational cost due to its supervised contrastive learning that processes two augmented views of each sample, and PHE also involves additional estimation and augmentation overhead.

# C. Broader Impact and Limitations Discussion

## C.1. Broader Impact.

Our method improves on-the-fly category discovery under open-world and streaming settings, which can benefit applications such as long-term perception, large-scale retrieval, and biodiversity monitoring by reducing the need for repeated manual re-labeling. At the same time, automatic discovery of novel categories on uncurated data streams may amplify existing dataset biases or unintentionally cluster sensitive attributes, so any real-world deployment should include careful data curation, monitoring, and human oversight.

## C.2. Limitations.

Our framework relies on strong pre-trained vision backbones (CLIP and DINOv2) and GPU resources, and its performance may degrade in low-resource scenarios or when such pre-training is unavailable. In addition, we mainly evaluate on standard OCD benchmarks with moderate numbers of novel classes, while more extreme non-stationary streams or very large numbers of emerging categories may increase prototype memory and adaptation instability.

## C.3. A Possible Solution in Future Work.

Future work could explore more lightweight or distilled backbones for deployment on edge devices, as well as stronger mechanisms for handling long-term distribution shift, e.g., memory-based replay or more robust prototype regularization. Another promising direction is to incorporate multimodal or human feedback to better name, merge, or filter discovered categories, improving both interpretability and safety in practical applications.