BATCHNORM DOMAIN ROUTING WITH LOW-RANK ADAPTERS FOR INCREMENTAL AUDIO CLASSIFICATION

Technical Report

Koichi Miyazaki, Katsuhiko Yamamoto

CyberAgent, Tokyo, Japan
{miyazaki_koichi_xa, yamamoto_katsuhiko}@cyberagent.co.jp

ABSTRACT

We describe our submitted system to DCASE 2026 Task 7, which studies domain-agnostic incremental audio classification. Our system keeps the shared convolutional neural network backbone frozen while adapting to new domains. For each new domain, it adds domain-specific low-rank adapters and classifier heads. At test time, it routes each clip to the branch whose batch normalization statistics best match the input. Combined with selective self-distillation, Batch normalization recalibration, and chunk-level prediction averaging, our best system achieves 70.1% average class-wise macro accuracy on the development test set, compared with 53.5% for the official baseline.

***Index Terms*—** incremental learning, domain adaptation, audio classification, low-rank adaptation, batch normalization

1. INTRODUCTION

DCASE 2026 Task 7 considers domain-agnostic incremental audio classification across three domains, D1, D2, and D3 [1]. The domains are introduced sequentially, while training data from earlier domains are unavailable and domain labels are not provided at test time. A system must therefore route each test audio to the correct domain branch before classification.

The official baseline uses a shared convolutional neural network (CNN) backbone [2], domain-specific batch normalization (BatchNorm) layers, and entropy-based domain selection [3]. However, entropy-based routing is often unreliable, and BatchNorm parameters alone provide limited capacity for new domains.

Several lines of prior work are related to this task. Knowledge distillation has been widely used in continual learning to preserve previously acquired knowledge without replay data [4]. Parameter-efficient adaptation methods such as residual adapters and low-rank adaptation (LoRA) introduce a small number of trainable parameters while keeping a shared backbone fixed [5, 6]. BatchNorm statistics have also been used to characterize domain-specific feature distributions and support domain generalization [7, 8]. These observations suggest that BatchNorm statistics may provide a more direct signal for domain identification than prediction entropy.

Based on these ideas, we improve domain routing by selecting the domain branch using BatchNorm statistics instead of prediction entropy. We further increase adaptation capacity by introducing domain-specific low-rank adapters and classifier heads while keeping the shared CNN backbone frozen. During incremental training, we apply selective self-distillation and BatchNorm recalibration to improve branch consistency and domain identification.

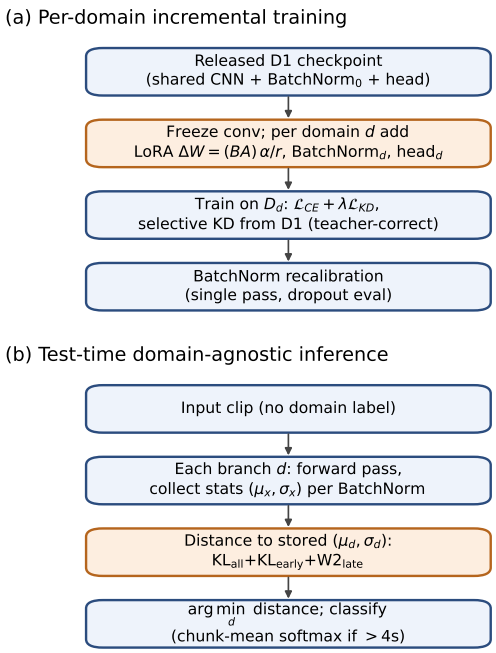

Figure 1: Proposed system. (a) We add LoRA-based low-rank adapters, BatchNorm layers, and classifier heads to a frozen backbone for a new domain d. (b) At test time, the branch whose BatchNorm statistics best match the clip is used for classification, described in Sec. 2.1.

Compared with the official baseline (53.5%), our best system achieves 70.1% average class-wise macro accuracy on the development test set. The remaining gap to oracle routing indicates that domain selection remains the primary challenge in this task.

2. PROPOSED METHOD

The proposed system extends the baseline by improving both domain routing and domain adaptation. The backbone is the baseline CNN, which consists of shared convolution layers and domain-specific BatchNorm layers. During incremental learning, we keep the shared backbone frozen and add lightweight domain-specific parameters for new domains. During inference, we replace entropy-based routing with BatchNorm-statistics-based routing. Figure 1 illustrates the overall training and inference procedures.

Algorithm 1 Domain-agnostic inference for one clip

Require: clip x; branches $d{=}0, \ldots, D{-}1$ with stored BN statistics (μ_d, σ_d)

for $d \leftarrow 0$ **to** $D-1$ **do**
  forward x through branch d; collect feature statistics (μ_x, σ_x) per BatchNorm layer
  $s_d \leftarrow$ compute distance to (μ_d, σ_d) ▷ Eq. (1)–(3)
end for
$\hat{d} \leftarrow \arg\min_d s_d$ ▷ select domain branch
if duration$(x) > 4$ s **then**
  $p \leftarrow$ mean softmax over 4 s chunks of branch $\hat{d}$
else
  $p \leftarrow$ softmax of branch $\hat{d}$
end if
return $\arg\max_c p_c$

2.1. BatchNorm-statistics domain routing

Domain selection is the main challenge in this task because domain labels are unavailable at test time. The official baseline selects a domain branch using prediction entropy. However, confidence-based criteria may be unreliable when multiple branches produce similar predictions.

BatchNorm running statistics (mean, standard deviation) capture domain-specific feature distributions accumulated during training. We therefore identify the domain branch whose stored BatchNorm statistics best match the input clip. For each branch d, we compare the BatchNorm statistics extracted from the input clip, denoted by (μ_x, σ_x), with the stored running statistics (μ_d, σ_d).

We compute both Wasserstein and Kullback-Leibler (KL) distances between the input and stored statistics:

$$s_d^{\mathrm{W2}} = \frac{1}{L}\sum_l \frac{1}{C_l}\sum_c \Big[(\mu_x - \mu_d)^2 + (\sigma_x - \sigma_d)^2\Big], \quad (1)$$

$$s_d^{\mathrm{KL}} = \frac{1}{L}\sum_l \frac{1}{C_l}\sum_c \mathrm{KL}\Big(\mathcal{N}(\mu_x, \sigma_x^2) \,\|\, \mathcal{N}(\mu_d, \sigma_d^2)\Big). \quad (2)$$

Our final routing score combines three normalized distances:

$$s_d = \tilde{s}_d^{\mathrm{KL,all}} + \tilde{s}_d^{\mathrm{KL,early}} + \tilde{s}_d^{\mathrm{W2,late}}, \quad (3)$$

where the three terms correspond to KL distance over all layers, KL distance over early layers, and Wasserstein distance over late layers. The branch with the lowest score is selected for classification. For clips longer than 4 s, we average predictions across non-overlapping chunks during inference. Algorithm 1 gives the full inference procedure.

2.2. Domain-specific adaptation

BN parameters alone provide limited capacity for adapting to new domains. To increase adaptation capacity while preserving the pretrained backbone, we introduce domain-specific low-rank adapters and classifier heads.

Each 3×3 convolution receives a low-rank update $\Delta W = (BA)\alpha/r$ with rank $r = 32$. The matrix B is initialized to zero so that the initial behavior matches the released model. We also add a domain-specific classifier head initialized from the corresponding pretrained checkpoint. The shared convolution layers remain frozen throughout training.

Table 1: Comparison of domain routing criteria using the released baseline checkpoints. D2 and D3 denote class-wise macro accuracy (%), and sel denotes domain-routing accuracy (%).

Selector	D2	D3	Avg	sel_2	sel_3
entropy (baseline)	59.2	47.7	53.5	23.0	2.1
energy	58.9	45.2	52.1	21.9	1.0
max-logit	58.9	45.5	52.2	22.4	0.9
bnstats_all	**64.6**	**48.8**	**56.7**	60.4	30.8
oracle	70.7	56.2	63.5	100	100

Table 2: Submitted systems on the D1-free development test set (class-wise macro accuracy, %). Systems 2–4 share the same adapted checkpoints and differ only in the selector.

#	Adaptation	Routing	D2	D3	Avg
–	Baseline	Entropy	59.2	47.7	53.5
1	Baseline	BatchNorm statistics	64.6	48.8	56.7
2	LoRA + KD	KL	71.8	62.4	67.1
3	LoRA + KD	Hybrid	73.6	63.7	68.7
4	LoRA + KD	BatchNorm statistics	73.5	66.7	**70.1**
–	LoRA + KD	Oracle	79.2	69.7	74.4

2.3. Incremental training

Incremental learning is challenging because data from earlier domains are unavailable during adaptation. This constraint can cause different branches to drift apart, making domain routing more difficult. To improve consistency across branches, we apply selective self-distillation.

We first adapt the D2 branch and then adapt the D3 branch. Each adaptation step uses only data from the current domain. The student branch is trained using cross-entropy loss together with knowledge distillation from the previous branch:

$$\mathcal{L}_{\mathrm{KD}} = \frac{T^2}{|\mathcal{C}|}\sum_{i\in\mathcal{C}} \mathrm{KL}\Big(\sigma\Big(z_i^{(d-1)}/T\Big) \,\|\, \sigma\Big(z_i^{(d)}/T\Big)\Big). \quad (4)$$

Unlike standard self-distillation, we apply distillation only to samples that are classified correctly by the teacher. This strategy reduces the risk of transferring incorrect predictions under domain shift.

After training each domain branch, we recompute BN running statistics using a single pass over the training data. Dropout is disabled during this recalibration step. The resulting statistics better reflect inference-time behavior and provide a cleaner signal for domain routing.

3. EXPERIMENTS

3.1. Experimental setup

The task consists of 10-class sound classification across three domains. We use the official D2 and D3 development datasets for training and evaluation. Audio is sampled at 32 kHz, and 64-dim log-mel features are computed within the model. Training uses 4

Table 3: Ablation study on the development test set. Each row adds one component to the configuration above. Results are reported as average class-wise macro accuracy (%).

Configuration	Avg	Δ
Baseline (entropy routing)	53.5	–
+ BatchNorm-statistics routing	56.7	+3.2
+ Domain-specific adaptation	64.4	+7.7
+ Higher-rank adapters and distillation	66.5	+2.1
+ Selective self-distillation	69.4	+3.0
+ D3 initialization from D2 branch	70.1	+0.7

s audio segments, while inference is performed on full clips. We report class-wise macro accuracy following the official evaluation protocol.

3.2. Evaluation of Domain Routing

We first evaluate whether BatchNorm statistics provide a better routing signal than confidence-based criteria. Table 1 compares entropy, energy, max-logit, and BatchNorm-statistics-based routing using the released baseline checkpoints. BatchNorm-statistics routing improves average accuracy from 53.5% to 56.7%, mainly through more accurate domain selection.

3.3. Main results

Table 2 compares the proposed method with the official baseline and several routing variants. Replacing entropy-based routing with BatchNorm-statistics routing consistently improves performance. Combining BatchNorm-statistics routing with LoRA adaptation, selective distillation, and BatchNorm recalibration yields the best result of 70.1.

3.4. Ablation study

Table 3 presents a cumulative ablation study. Adding domain-specific adaptation capacity provides the largest improvement, increasing average accuracy by 7.7 percentage points. Selective self-distillation further improves performance by 3.0 points. Initializing D3 from the trained D2 branch gives a smaller but consistent gain.

4. DISCUSSION

Domain Routing Remains the Primary Bottleneck: The experimental results suggest that domain routing is the main source of error in this task. Although the proposed system improves average accuracy from 53.5% to 70.1%, oracle routing achieves 74.4% using the same adapted checkpoints. This gap suggests that the adapted classifiers are substantially stronger than the routing mechanism.

The results also show that improvements in adaptation capacity alone are insufficient. Domain-specific LoRA adapters and selective distillation improve branch specialization, but their benefits depend on selecting the correct branch at inference time. Further improvements are therefore likely to come from more accurate domain routing rather than larger adaptation modules.

Limitations and Future Directions: A limitation of the development setup is that D1 samples are not included in the released development test set. As a result, routing performance for D1 cannot be measured directly during development. Consequently, we submitted multiple systems (#1–#4) with different routing strategies, including the best-performing system (#4).

Another limitation is that BatchNorm statistics may capture factors beyond semantic content. For example, training segments are zero-padded to a fixed duration, which may introduce length-related cues into the stored statistics. A better understanding of the information encoded in BatchNorm statistics could help develop more robust routing methods.

Future work will investigate routing methods that combine BatchNorm statistics with learned domain representations. Reducing the remaining routing error appears to be the most promising direction for improving domain-agnostic incremental audio classification.

5. CONCLUSION

We presented a domain-agnostic incremental audio classification system for DCASE 2026 Task 7. The proposed method combines BatchNorm-statistics-based domain routing with domain-specific LoRA adapters, selective self-distillation, and BatchNorm recalibration while keeping the shared CNN backbone frozen. On the development test set, the proposed system improves average class-wise macro accuracy from 53.5% for the official baseline to 70.1%.

Our results suggest that domain routing is a more significant challenge than adaptation capacity in this task. Although the proposed adaptation strategy substantially improves classification performance, a noticeable gap remains between the best routing strategy and oracle routing. Future work should therefore focus on more accurate and robust domain identification methods for domain-agnostic incremental learning.

6. REFERENCES

[1] R. Casciotti, M. Mulimani, M. Harju, J. R. Jensen, and A. Mesaros, "Domain-agnostic incremental learning for sound classification: A DCASE 2026 challenge task," arXiv:2606.02173, 2026.

[2] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[3] M. Mulimani and A. Mesaros, "Domain-incremental learning for audio classification," in *Proc. ICASSP*, 2025.

[4] Z. Li and D. Hoiem, "Learning without forgetting," in *Proc. ECCV*, 2016, pp. 614–629.

[5] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Learning multiple visual domains with residual adapters," in *Proc. NeurIPS*, 2017.

[6] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *Proc. ICLR*, 2022.

[7] M. Segu, A. Tonioni, and F. Tombari, "Batch normalization embeddings for deep domain generalization," *Pattern Recognition*, vol. 135, pp. 109–115, 2023.

[8] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell, "TENT: Fully test-time adaptation by entropy minimization," in *Proc. ICLR*, 2021.