A DOMAIN-AGNOSTIC INCREMENTAL LEARNING FOR AUDIO CLASSIFICATION SYSTEM FOR DCASE 2026 TASK 7

Technical Report

Peiwei Chang, Yuelan Cheng, Yongqiang Chen, Philip J.B. Jackson, Wenwu Wang

Centre for Vision, Speech, and Signal Processing (CVSSP)
University of Surrey
Guildford, UK

ABSTRACT

We describe our submissions to DCASE2026 Task 7 (Domain-Agnostic Incremental Learning for Audio Classification). Starting from the provided CNN14 baseline frozen at its D1 checkpoint, each incremental domain learns a strictly disjoint set of parameters, including domain-specific batch normalization, low-rank (r=8) convolutional adapters, and a zero-initialized per-domain classifier delta. Catastrophic forgetting is therefore prevented by construction. At inference time, a small residual-MLP ensemble routes between domain heads using multi-scale pooled features extracted from the frozen D1 path. The final logits are computed as a router-weighted mixture of per-domain ensemble heads, whose member subsets are selected independently for each head by exhaustive screening over 28 trained variants. On the development test set our primary system achieves 77.5% average accuracy (D2: 85.7%, D3: 69.3%).

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

1. INTRODUCTION

Task 7 of the DCASE2026 Challenge [1] requires learning sound-event classification over three domains $D1 \rightarrow D2 \rightarrow D3$ sequentially, with no access to previous domains' data at each incremental step, and with external data and pretrained models prohibited. Only a trained model checkpoint is provided for D1; development audio is available for D2 (139 min) and D3 (275 min) over ten sound classes. The baseline system [2] adapts domain-specific batch-normalization (BN) layers per domain and selects a domain at test time by minimum prediction entropy.

Our work keeps the baseline's domain-specific components structure but addresses three bottlenecks. First, BN-only adaptation ($\sim$30 k parameters per domain) underfits the new domains, we therefore add low-rank convolutional adapters [3, 4] and a per-domain classifier delta. Second, entropy-based domain selection is unreliable, in our measurements it routes only 16% of D3 test clips to the D3 head, because the D1 head is systematically overconfident. We replace it with a learned domain router. Third, individual models are noisy under the small data budget, we therefore train many domain-parameter variants on a shared frozen backbone and select ensemble member subsets for each domain head. The full system is summarized in Fig. 1.

2. SYSTEM DESCRIPTION

2.1. Architecture and per-domain parameters

The backbone is the provided MCnn14 network, a CNN14 [5] variant with per-domain BN layers [6, 2]. A 4 s clip is converted to a $1\times400\times64$ log-mel spectrogram (channel$\times$time$\times$frequency) and passed through six convolutional blocks. Each block applies two 3×3 convolutions, each followed by per-domain batch normalization and ReLU, then 2×2 average pooling; the channel width doubles every block ($64\rightarrow128\rightarrow256\rightarrow512\rightarrow1024\rightarrow2048$) while the time–frequency resolution halves, yielding a $2048\times6\times1$ feature map. Global mean+max pooling produces a 2048-d embedding that a linear classifier maps to the ten classes. We freeze every shared weight at the provided D1 checkpoint and attach three groups of domain-specific parameters per incremental domain d:

(a) Batch normalization. Affine parameters and running statistics of all BN layers, as in the baseline.

(b) Low-rank convolutional adapters. For each convolution W with input x, the adapted output is $Wx + B_d A_d x$, where A_d is a 3×3 convolution to rank r=8 and B_d a 1×1 convolution back to the output channels. B_d is zero-initialized, so each adapter exactly matches the frozen backbone at initialization. This is the convolutional analogue of LoRA [3] and closely related to residual adapters [4].

(c) Classifier delta. The baseline freezes the D1 classifier for all domains, which types the decision boundary of new domains to D1. We add a zero-initialized domain-specific linear layer whose output is added to the frozen classifier: $z = W_{fc}\,h + W_{fc,d}\,h$.

Together, these adds $\approx$0.53 M trainable parameters (0.7% of the 77.1 M backbone). Because domains share no trainable parameters, training D3 cannot overwrite D2 or D1. Forgetting is prevented by construction at the parameter level, rather than through regularization, and the method extends to future domains with constant per-domain cost.

2.2. Incremental training

Training recordings are split into non-overlapping 4 s segments (Table 1). Each incremental step trains only the new domain's parameters using cross-entropy on that domain's chunks. We apply SpecAugment [7] to the log-mel features and waveform augmentation including gain jitter, additive Gaussian noise, and circular time shifting. Pitch shifting and time stretching *reduced* accuracy in our experiments. For several target classes (piano, alarm, telephone ringing), pitch and tempo are part of the class identity, unlike

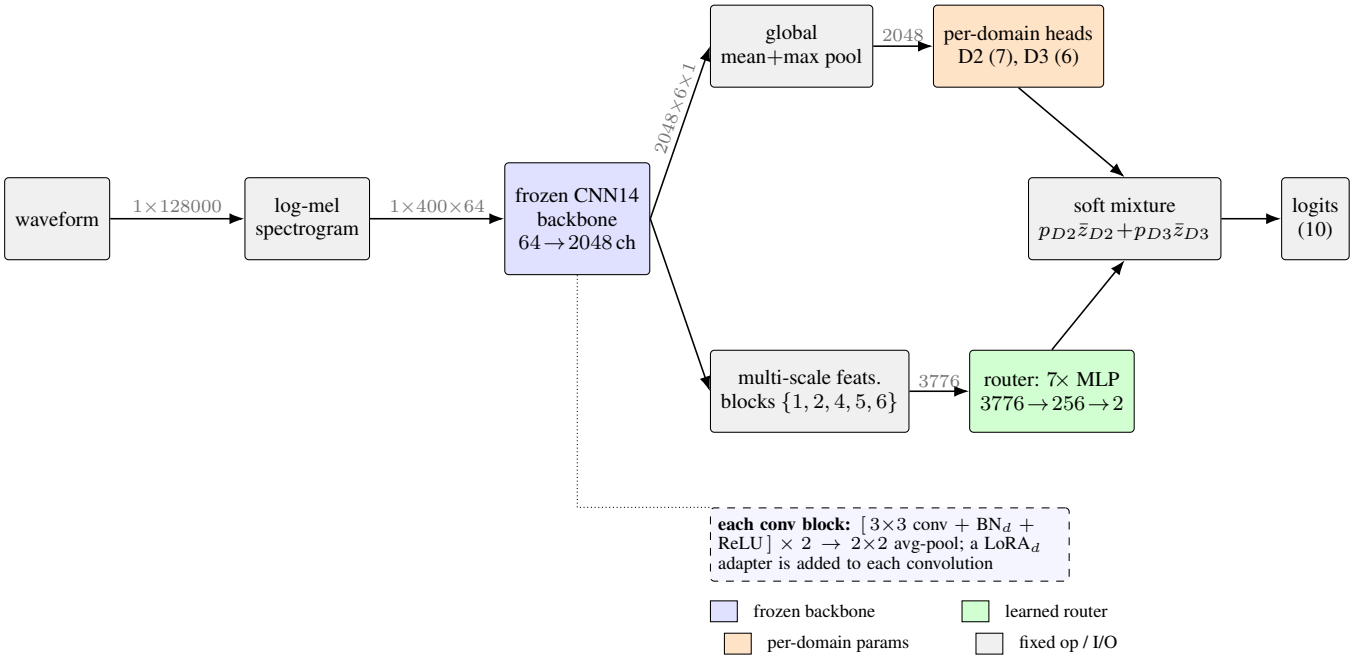

Figure 1: System overview. The classification path (top: backbone → global pool → per-domain heads) and the routing path (bottom: multi-scale features → router) merge at the soft mixture. The CNN14 backbone is trained on D1 then frozen and shared. Per-domain parameters (BN, LoRA adapters, classifier delta) are trained separately for each domain. The router is trained to discriminate D2 from D3. The router uses only frozen-path (D1) features.

Table 1: Training and routing hyperparameters.

Input	32 kHz mono, 4 s chunks
Features	log-mel, 64 bands, 50–14 k Hz, win 1024, hop 320
Optimizer	Adam, batch 128, 120 epochs / domain
LR schedule	cosine $10^{-3} \rightarrow 10^{-5}$
LoRA	rank 8 on all 3×3 convolutions
SpecAugment	2×32 time, 2×8 frequency masks
Waveform augm.	gain ±6 dB; noise 15–30 dB SNR; shift ±0.5 s
Router	7× residual MLP, $3776 \rightarrow 256 \rightarrow 2$
Router training	AdamW 10^{-3}, wd 10^{-4}, 60 epochs, early stop on a held-out 10% of train chunks

in speech tasks where these augmentations are often safe.

We trained 28 variants of the domain parameters on the same frozen backbone using three recipes: 22 random seeds of the standard recipe, 3 seeds with the classifier delta enabled, and 3 seeds in which D3's adapters and BN are warm-started from the trained D2 parameters instead of from zero (motivated by D2 and D3 both being real-world re-recordings of the clean D1 source material). Each variant costs only the ∼0.53 M domain parameters; the backbone is stored once.

2.3. Learned domain routing

At test time the domain label is unknown. The baseline selects the domain whose head gives the lowest prediction entropy. In our measurements, this rule routes only 16% of D3 clips to the D3 head (the D1 head is over-confident). This caps the baseline's average accuracy well below what its domain-specific heads can otherwise achieve.

We instead train a small domain classifier, which is an ensemble of seven residual MLPs ($3776 \rightarrow 256$, one residual block, different seeds, averaged in probability space) that discriminates D2 from D3 chunks. Its input is the concatenation of mean+max-pooled outputs of convolution blocks $\{1, 2, 4, 5, 6\}$, computed along the frozen D1 path (all per-domain components set to D1). This design led to three empirical findings:

Multi-scale helps. Block-{1,2,4,5,6} concatenation outperforms the final-block embedding by $+0.9$ points in average accuracy. Adding block 3 consistently reduced performance in all combinations we tested.

Only the frozen path generalizes. Routing features (or head logits) extracted from the adapted D2/D3 paths degrade routing accuracy by 0.4–1.2 points. The adapted paths overfit their training chunks, so router inputs computed on training data do not match test-time statistics. In contrast, The frozen D1 path has never been updated to D2/D3 data and transfers more cleanly.

Soft mixing beats hard selection. Final logits are the mixture

$$z \;=\; p_{D2}\,\bar{z}_{D2} \;+\; p_{D3}\,\bar{z}_{D3}, \tag{1}$$

where (p_{D2}, p_{D3}) is the router posterior and $\bar{z}_D$ denotes the logit average of domain D's ensemble head. Sharpening or hardening the posterior reduced accuracy. Fig. 2(b) shows the posterior is strongly bimodal, with a small mass of genuinely ambiguous clips for which the soft mixture provides a hedge.

2.4. Ensembling and per-head member selection

Each domain head averages logits over a subset of the 28 trained variants. Since the two heads enter (1) independently, their member

Table 2: Submitted systems. Params in millions; GMACs approximate, per 4 s clip.

#	Members	Heads	Routing	Par.	GMACs
1	11 (3 recipes)	7 / 6 split	learned soft	96.7	∼119
2	4 (standard)	4 / 4 joint	learned soft	89.0	∼77
3	3 (standard)	3 / 3 joint	hybrid, τ=0.6	88.0	∼60
4	1 (standard)	single	temp. + entropy	78.2	∼26

subsets need not be equal. We therefore select the D2-head subset and the D3-head subset separately. Starting from the best joint subset found by exhaustive search over all subsets of size ≤ 8, we perform coordinate ascent by alternately sweeping one head's subset exhaustively while holding the other fixed, using development-test average accuracy as the objective. The search converges in two iterations.

The optimal D2 head uses seven members, dominated by classifier-delta variants, whereas the optimal D3 head uses six members of the standard recipe and one warm-started member. Split-head selection, classifier-delta members, and warm-started members each contribute small additional gains over the best joint subset. Notably, members that are individually weak can still enter the optimal subset by contributing decorrelated errors, while adding more members beyond the optimum consistently reduces performance.

3. SUBMITTED SYSTEMS

Table 2 summarizes the four submissions; all use the same frozen backbone and training recipes, differing in ensemble size and routing.

Submission 1 (primary) is the full system. Eleven members spanning all three training recipes, split per-head subsets (7 for D2, 6 for D3), and learned soft routing.

Submission 2 is a simpler, more conservative variant. Four standard-recipe members, the same subset used for both heads, learned soft routing. It isolates the contribution of the per-head and multi-recipe machinery (about -0.9 points relative to the primary).

Submission 3 addresses a protocol risk. The learned router only ever predicts D2 or D3, so D1-like evaluation content would never reach the D1 head. Here, whenever router confidence $\max(p_{D2}, p_{D3}) < 0.6$, the system falls back to entropy-minimum selection over all three heads, keeping the D1 head reachable at a measured cost of $\sim$1 point on the D2/D3-only development test.

Submission 4 is a minimal single-model backup without a learned router. Per-domain logits of one standard member are temperature-scaled with $T = [8, 1, 1]$ (flattening the over-confident D1 head) and the lowest-entropy domain is selected. This is the simplest system that fixes the baseline's routing failure, at 78.2 M parameters and ∼26 GMACs.

4. RESULTS

Table 3 reports step-wise results of the official baseline [2] and our four submitted systems. The baseline reaches 58.6% on D2 after step D2, and 59.0%/46.1% on D2/D3 after step D3 (52.55% average). Our primary system raises the average by roughly 25 points to 77.52%, with the largest gains coming from the low-rank adapter capacity and the learned domain router, and further smaller gains from per-head ensemble selection and the classifier delta. No system loses D2 accuracy across the D3 step, so forgetting is zero by construction.

Table 3: Step-wise development-test accuracy (%) of the official baseline [2] and our four submitted systems.

	after D2	after D3		
#	D2	D2	D3	Avg.
Base.	58.60	59.00	46.10	52.55
1	84.86	85.74	69.30	**77.52**
2	83.04	84.29	68.91	76.60
3	83.03	83.15	70.31	76.73
4	82.87	82.11	59.48	70.80

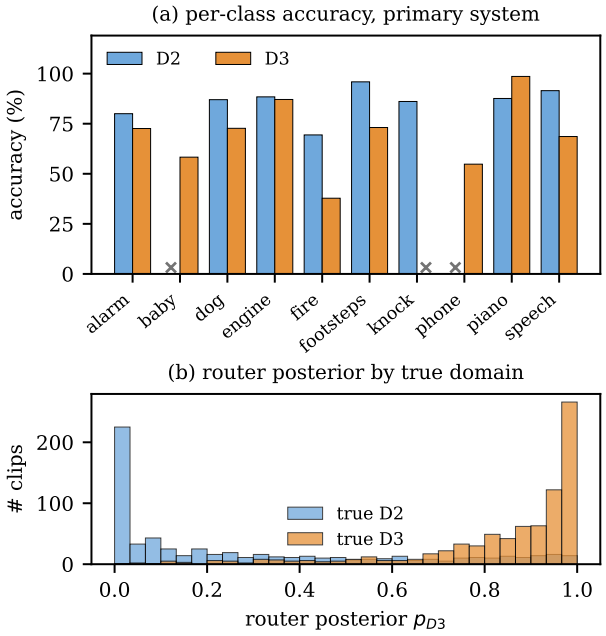

Figure 2: Primary system on the development test set: (a) per-class accuracy by domain; gray crosses mark class–domain pairs that do not occur in the dataset (D2 contains no baby/phone clips, D3 no knock clips); (b) router posterior p_{D3} for clips of each true domain.

Fig. 2(a) shows the remaining errors concentrate in D3, particularly the acoustically overlapping classes (fire 37.8%, phone 54.8%, baby 58.3%, speech 68.6%) recorded under low signal-to-noise conditions. In contrast, D2 per-class accuracy is consistently high (69–96%). Fig. 2(b) confirms the learned router separates the domains with a clear bimodal posterior.

5. CONCLUSION

We address domain-incremental sound classification under the strict Task 7 protocol, where previous-domain data, external data, and pretrained models are unavailable, while requiring that learning a new domain does not degrade performance on earlier ones. Our approach freezes a shared backbone and attaches strictly disjoint per-domain parameters, including domain-specific batch normalization, low-rank adapters, and a classifier delta, so that forgetting is

prevented by construction. To address the baseline's main weakness, we replace entropy-based domain selection with a lightweight learned router and apply per-head ensemble selection, aiming to close the gap between the baseline's strong per-domain heads and its weak routing mechanism. On the development set, the proposed system improves average accuracy from 52.6% to 77.5%, while preserving earlier-domain accuracy throughout the incremental sequence.

The most interesting finding is that the dominant bottleneck was not representation quality, but domain routing. The per-domain heads were already strong, and replacing entropy-based selection with a classifier trained on frozen-path features recovered most of the lost performance. In contrast, routing signals extracted from the adapted paths were ineffective, because those paths specialize to, and partially memorize, their own training domains. This suggests that future work on domain-incremental audio should treat routing and selection as first-class components, for example through calibration-aware or distribution-based routers, and should report routing accuracy separately from per-domain head accuracy to clarify where improvements originate. Since the additional parameter cost is constant, approximately (0.5)M parameters per domain, the method also extends naturally to longer domain sequences.

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] M. Mulimani and A. Mesaros, "Domain-incremental learning for audio classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.

[3] 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 *International Conference on Learning Representations (ICLR)*, 2022.

[4] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Learning multiple visual domains with residual adapters," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[5] 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.

[6] Y. Li, N. Wang, J. Shi, X. Hou, and J. Liu, "Adaptive batch normalization for practical domain adaptation," *Pattern Recognition*, vol. 80, pp. 109–117, 2018.

[7] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019, pp. 2613–2617.