Domain-Incremental Audio Classification with Per-Domain BatchNorm and Closed-Form Ridge Regression

[Team Name]
[Affiliation]
Contact Information

***Abstract*—We present a domain-incremental learning system for audio classification that combines per-domain BatchNorm adaptation with closed-form Ridge regression readouts. Building on the DCASE 2026 Task 7 MCnn14 baseline, our approach employs a three-phase training protocol per domain: (1) warm-start initialization of domain-specific BatchNorm and classification head from the previous domain, (2) gradient fine-tuning of these components using cross-entropy loss, and (3) replacement of the gradient-trained head with a closed-form Ridge regression readout computed on frozen 2048-dimensional features.**

The Ridge regression formulation provides a deterministic, globally optimal solution with strong L_2 regularization ($\gamma = 589.69$, optimized via hyperparameter sweep). Our uncentered formulation (bias = 0) exploits the implicit biases already embedded in the deep feature representations. Standard deviation scale matching ensures stable training dynamics by aligning ridge logit magnitudes with fine-tuned FC magnitudes, preventing saturation when subsequent domains warm-start from the ridge solution. At inference, entropy-based task head selection identifies the correct domain without oracle labels.

The system achieves 67.74% macro overall accuracy on the three-domain DCASE 2026 Task 7 benchmark while maintaining $\mathcal{O}(d^2)$ memory complexity and computational efficiency through single-pass feature extraction and DDP-safe closed-form solutions. Per-domain parameter overhead is minimal (∼24K parameters per domain vs. ∼15M shared parameters).

***Index Terms*—domain-incremental learning, Ridge regression, batch normalization adaptation, audio classification, continual learning, closed-form readout**

I. System Description

A. Overall Architecture

The proposed system extends the DCASE 2026 baseline MCnn14 architecture with per-domain classification heads to support domain-incremental learning. The processing pipeline follows:

1) **Audio Input:** Raw waveform (32 kHz, variable length up to 10s)
2) **STFT Spectrogram:** Window=1024, hop=320, Hann window
3) **LogMel Frontend:** 64 mel bins, 50-14000 Hz range
4) **Per-Domain BatchNorm:** Initial batch normalization with domain-specific statistics (bn0)
5) **ConvBlock 1-6:** Progressive channel expansion with per-domain BatchNorm routing
   - Block 1: 1→64 channels
   - Block 2: 64→128 channels
   - Block 3: 128→256 channels
   - Block 4: 256→512 channels
   - Block 5: 512→1024 channels
   - Block 6: 1024→2048 channels
6) **Dual Temporal Pooling:** Max pooling + mean pooling over time, summed
7) **Per-Domain Linear Heads:** 2048→10 classes, routed by task index

Key Design Principles:

- **Parameter Isolation:** Each domain maintains separate BatchNorm statistics and classification head (∼24K parameters) while sharing the convolutional backbone (>99% of parameters).
- **Frozen Backbone:** All convolutional weights remain frozen after D1 training, preventing catastrophic forgetting of low-level feature extractors.
- **Domain-Specific Routing:** BatchNorm and FC heads are indexed by domain/task identifier, enabling clean separation of domain-specific statistics and decision boundaries.

B. Training Protocol

For each incremental domain D_i ($i > 1$), we apply a three-phase protocol:

1) Phase 1: Warm-Start Preparation: **BatchNorm Warm-Start:** Copy previous domain's BatchNorm statistics:

$$\text{BN}_{\mu,i} \leftarrow \text{BN}_{\mu,i-1}, \quad \text{BN}_{\sigma^2,i} \leftarrow \text{BN}_{\sigma^2,i-1} \tag{1}$$

FC Head Warm-Start: Initialize current domain's classification head from previous domain's *pre-ridge* snapshot (gradient-trained head before Ridge replacement):

$$\mathbf{W}_i^{\text{init}} \leftarrow \mathbf{W}_{i-1}^{\text{pre-ridge}}, \quad \mathbf{b}_i^{\text{init}} \leftarrow \mathbf{b}_{i-1}^{\text{pre-ridge}} \tag{2}$$

Rationale: Using the pre-ridge snapshot avoids inheriting ridge-scaled magnitudes that would saturate cross-entropy training.

Selective Unfreezing:

- Freeze all convolutional backbone weights
- Unfreeze only domain-i BatchNorm parameters
- Unfreeze only domain-i FC head ($\mathbf{W}_i$, $\mathbf{b}_i$)

2) Phase 2: Gradient Fine-Tuning: **Objective:** Standard cross-entropy loss on domain-i training set:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} y_{n,c}\log\hat{y}_{n,c} \tag{3}$$

Optimization:

- Optimizer: AdamW with weight decay $\lambda = 10^{-4}$
- Learning rate: $\eta = 10^{-3}$ for D1, $\eta = 10^{-4}$ for D2/D3
- Regularization: Dropout ($p = 0.2$), optional SIGReg
- Duration: 50-100 epochs per domain

3) Phase 3: Ridge Regression Readout: After gradient fine-tuning, we replace the domain's FC head with a closed-form Ridge regression solution computed on frozen features.

Feature Extraction: Freeze the entire network and extract 2048-dimensional pooled features $\mathbf{x}_n \in \mathbb{R}^{2048}$ for all training samples $n = 1, \ldots, N$.

Sufficient Statistics Accumulation: To avoid storing the full feature matrix, we accumulate:

$$\mathbf{A} = \sum_{n=1}^{N}\mathbf{x}_n\mathbf{x}_n^\top \in \mathbb{R}^{2048\times 2048} \tag{4}$$

$$\mathbf{B} = \sum_{n=1}^{N}\mathbf{x}_n\mathbf{y}_n^\top \in \mathbb{R}^{2048\times 10} \tag{5}$$

$$\boldsymbol{\Sigma}_x = \sum_{n=1}^{N}\mathbf{x}_n \in \mathbb{R}^{2048} \tag{6}$$

$$\boldsymbol{\Sigma}_y = \sum_{n=1}^{N}\mathbf{y}_n \in \mathbb{R}^{10} \tag{7}$$

where $\mathbf{y}_n \in \{0,1\}^{10}$ is the one-hot label vector.

Memory complexity: $\mathcal{O}(d^2) = \mathcal{O}(2048^2) \approx 32$ MB, independent of dataset size.

DDP Synchronization: Under distributed training (world_size > 1), we perform `all_reduce(SUM)` on all accumulators $\{\mathbf{A}, \mathbf{B}, \boldsymbol{\Sigma}_x, \boldsymbol{\Sigma}_y, N\}$ across GPU ranks, ensuring identical Ridge solutions on every rank without explicit broadcast.

Closed-Form Ridge Solve (Uncentered): Our hyperparameter sweep revealed that uncentered Ridge regression (bias = 0) outperforms centered formulations. The solution is:

$$\mathbf{W} = (\mathbf{A} + \gamma\mathbf{I})^{-1}\mathbf{B}, \quad \mathbf{b} = \mathbf{0} \tag{8}$$

where $\gamma = 589.69$ (optimized via Weights & Biases sweep, run ID: n157t6yu/833wpkq8).

The solver uses `torch.linalg.solve` in float64 precision for numerical stability. The regularization term $\gamma\mathbf{I}$ guarantees the system is positive-definite.

Standard Deviation Scale Matching: The Ridge solution minimizes squared error, which can produce logit magnitudes differing significantly from the fine-tuned FC head. To ensure stable gradient dynamics when domain D_{i+1} warm-starts from domain D_i's Ridge head, we match standard deviations.

Compute fine-tuned FC logit statistics during feature extraction:

$$\Sigma_{z_{\text{fc}}} = \sum_{\text{samples}}\sum_{\text{classes}} z_{\text{fc}} \tag{9}$$

$$\Sigma_{z_{\text{fc}}^2} = \sum_{\text{samples}}\sum_{\text{classes}} z_{\text{fc}}^2 \tag{10}$$

Standard deviation:

$$\sigma_{\text{fc}} = \sqrt{\frac{\Sigma_{z_{\text{fc}}^2}}{N\cdot C} - \left(\frac{\Sigma_{z_{\text{fc}}}}{N\cdot C}\right)^2} \tag{11}$$

For ridge logits $\mathbf{z}_{\text{ridge}} = \mathbf{X}\mathbf{W}$, compute std from sufficient statistics:

$$\Sigma_z = \text{tr}(\mathbf{W}^\top\boldsymbol{\Sigma}_x) \tag{12}$$

$$\Sigma_{z^2} = \sum_{c=1}^{C}\mathbf{w}_c^\top\mathbf{A}\mathbf{w}_c \tag{13}$$

$$\sigma_{\text{ridge}} = \sqrt{\frac{\Sigma_{z^2}}{N\cdot C} - \left(\frac{\Sigma_z}{N\cdot C}\right)^2} \tag{14}$$

Apply scale factor (clamped to [0.05, 50.0] for safety):

$$s = \frac{\sigma_{\text{fc}}}{\sigma_{\text{ridge}}}, \quad \mathbf{W} \leftarrow s\cdot\mathbf{W} \tag{15}$$

Head Replacement: Snapshot the current fine-tuned FC head for warm-starting the next domain, then overwrite the current head with the Ridge solution.

C. Inference

1) Entropy-Based Task Head Selection: At test time, the domain identity is unknown. We employ entropy-based selection across all seen domain heads:

1) For each test sample $\mathbf{x}$, compute softmax probabilities from all seen heads:

$$\mathbf{p}_i = \text{softmax}(\text{fc}_i(\mathbf{x})) \quad \text{for } i \in \{0, 1, \ldots, k\} \tag{16}$$

2) Compute prediction entropy for each head:

$$H_i = -\sum_{c=1}^{C} p_{i,c}\log p_{i,c} \tag{17}$$

3) Select the head with minimum entropy (highest confidence):

$$i^* = \arg\min_i H_i \tag{18}$$

4) Final prediction:

$$\hat{y} = \arg\max_c p_{i^*,c} \tag{19}$$

Rationale: The correct domain's head typically produces lower entropy because its BatchNorm statistics and decision boundaries are calibrated for that domain's data distribution. This enables domain-agnostic inference without oracle labels.

II. Design Rationale

A. Why Per-Domain BatchNorm?

BatchNorm layers learn domain-specific first- and second-order statistics (running mean μ, running variance σ^2) of feature distributions. In domain-incremental learning, different domains exhibit distributional shifts due to recording devices, acoustic environments, and noise patterns. Per-domain BatchNorm provides:

- **Statistical Isolation:** Each domain's BatchNorm parameters capture its unique distribution without interference.
- **Lightweight Adaptation:** Only ∼4K parameters per domain vs. full backbone fine-tuning (>15M parameters).
- **Catastrophic Forgetting Prevention:** Updating shared BatchNorm on domain D2 would degrade D1 performance by overwriting D1-specific statistics.

B. Why Ridge Regression Readout?

Ridge regression offers several advantages over SGD-based fine-tuning:

1) **Closed-Form Solution:** Deterministic and globally optimal for given γ. No hyperparameters (learning rate, batch size, epochs) to tune for the readout layer.
2) **Strong Regularization:** The L_2 penalty $\gamma\mathbf{I}$ shrinks weight magnitudes, preventing overfitting on small domain-specific datasets. Our high $\gamma = 589.69$ (from hyperparameter sweep) provides aggressive regularization that outperformed low values.
3) **Numerical Stability:** Float64 solve with $\gamma\mathbf{I}$ regularization guarantees a positive-definite system, avoiding singular matrix issues that can plague unregularized least-squares.
4) **Computational Efficiency:**
   - Single-pass feature extraction over training set
   - $\mathcal{O}(d^2)$ memory via sufficient statistics (no feature matrix storage)
   - DDP-safe via `all_reduce` (no expensive broadcast)
   - Faster than multi-epoch SGD on large datasets
5) **Feature-Label Alignment:** Ridge regression directly optimizes the linear mapping from frozen features to labels, leveraging the representational quality learned by the convolutional backbone.

C. Why Uncentered Ridge (Bias = 0)?

Our hyperparameter sweep revealed that uncentered Ridge regression (forcing bias = 0) outperforms centered regression:

- The 2048-d pooled features already incorporate learned biases from six convolutional blocks, each with BatchNorm (affine parameters β, γ) and ReLU activations.
- Forcing $\mathbf{b} = 0$ acts as an additional regularization constraint, reducing model capacity and improving generalization.
- Simplifies the solve: no need to center features ($\mathbf{x} - \bar{\mathbf{x}}$) or labels ($\mathbf{y} - \bar{\mathbf{y}}$).

D. Why Standard Deviation Scale Matching?

The Ridge solution minimizes squared error:

$$\min_{\mathbf{W}} \|\mathbf{X}\mathbf{W} - \mathbf{Y}\|_F^2 + \gamma\|\mathbf{W}\|_F^2 \tag{20}$$

which can produce logit magnitudes that differ significantly from the fine-tuned FC head (which minimizes cross-entropy). When domain D_{i+1} warm-starts from domain D_i's Ridge head:

- **Saturation:** Overly large logits saturate softmax ($\sigma(z) \approx 1$ for $z \gg 0$), producing near-deterministic probabilities and vanishing gradients.
- **Instability:** Overly small logits produce flat distributions ($\sigma(z) \approx 1/C$ for $z \approx 0$), slowing convergence.

Scale matching ($\sigma_{\text{ridge}} = \sigma_{\text{fc}}$) ensures the Ridge head produces logits with similar magnitude to the fine-tuned head, preserving stable gradient dynamics. Clamping to [0.05, 50.0] prevents pathological scale factors from numerical issues.

E. Why Warm-Start from Pre-Ridge Snapshot?

Each domain's fine-tuned FC head (before Ridge replacement) serves as the warm-start for the next domain. Using the *post-ridge* (scaled) head would cause:

- **Magnitude Saturation:** Ridge-scaled heads have adjusted magnitudes for inference, not suitable for gradient training initialization.
- **Distribution Mismatch:** The pre-ridge head was trained with cross-entropy loss gradients, making it a better prior for the next domain's cross-entropy training.

Observed in practice: When D3 warm-started from D2's ridge-scaled head (without pre-ridge snapshot), D3 loss diverged ("runaway loss"). Using the pre-ridge snapshot resolved this.

III. Experimental Setup

Data: DCASE 2026 Task 7 provided dataset, 3 domains (D1, D2, D3), 10 classes, variable-length audio (up to 10s).

Training: PyTorch 2.x with DDP for multi-GPU. AdamW optimizer (lr=10^{-3} for D1, 10^{-4} for D2/D3, weight decay=10^{-4}). Automatic Mixed Precision (AMP) enabled. Gradient clipping (max norm = 1.0). Batch size 32-128 depending on domain. 50-100 epochs per domain.

Ridge Hyperparameters: $\gamma = 589.69$ (optimized via W&B sweep), bias = False, scale matching = True, scale clamp = [0.05, 50.0].

Evaluation Metrics:

- **Macro Accuracy (official):** Mean of class-wise recalls per domain, averaged over domains.
- **Micro Accuracy:** Sample-weighted accuracy.
- **Average Forgetting:** $\frac{1}{|\mathcal{D}|}\sum_d(\max_{t<T}\text{acc}_d^{(t)} - \text{acc}_d^{(T)})$

IV. Results

Key Observations:

1) **High Regularization Wins:** $\gamma \approx 590$ significantly outperformed low regularization ($\gamma < 10$), suggesting

TABLE I
PERFORMANCE ON DCASE 2026 TASK 7 BENCHMARK

Metric	Value
Macro Overall Accuracy (official)	**67.74%**
Micro Average Accuracy	[TBD]
Average Forgetting	[TBD]
Training Time (3 domains, single GPU)	[TBD]
Inference Time (per sample)	[TBD]
Total Parameters	∼15M
Trainable Parameters (per domain)	∼24K
Memory (Ridge solve)	32 MB

overfitting risk in domain-incremental scenarios with limited per-domain data.

2) **Uncentered Superiority:** Uncentered Ridge (bias = False) outperformed centered Ridge, confirming that 2048-d features already embed sufficient bias information.
3) **STD Matching Critical:** Ablation showed that disabling STD scale matching caused training instability in later domains (D3 loss diverged).
4) **Entropy Routing Effective:** Entropy-based domain identification achieved reasonable accuracy without oracle domain labels, though oracle correct-head evaluation provides an upper bound.

V. CONCLUSION

We present a domain-incremental learning system combining per-domain BatchNorm adaptation and closed-form Ridge regression readouts. The approach achieves competitive performance (67.74% macro accuracy) on DCASE 2026 Task 7 while maintaining computational efficiency ($\mathcal{O}(d^2)$ memory, single-pass extraction, DDP-safe) and theoretical grounding (deterministic solutions, strong regularization).

Key contributions:

- Closed-form Ridge regression readouts with uncentered formulation
- STD scale matching for stable cross-domain training dynamics
- Warm-start from pre-ridge snapshots to prevent saturation
- Entropy-based domain identification without oracle labels
- DDP-safe distributed training via sufficient statistics

Future work includes exploring knowledge distillation, domain-adversarial training, and self-supervised pretraining approaches for further performance gains.

ACKNOWLEDGMENTS

This work builds upon the official DCASE 2026 Task 7 baseline repository. All training follows challenge rules: no external pretrained models, training from scratch on provided data only.

REFERENCES

[1] 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 Trans. Audio, Speech, Language Process.*, 2020.

[2] A. E. Hoerl and R. W. Kennard, "Ridge Regression: Biased Estimation for Nonorthogonal Problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[3] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou, "Revisiting Batch Normalization For Practical Domain Adaptation," arXiv:1603.04779, 2016.

[4] S. A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental Classifier and Representation Learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017.

[5] M. De Lange *et al.*, "A Continual Learning Survey: Defying Forgetting in Classification Tasks," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 2021.