Journal of Computer Science and Electrical Engineering
Print ISSN: 2663-1938
Online ISSN: 2663-1946
DOI: https://doi.org/10.61784/jcsee4152

CLASS-CONDITIONAL INTERMEDIATE-DOMAIN ADVERSARIAL ADAPTATION FOR CROSS-DEVICE ACOUSTIC SCENE CLASSIFICATION

ZiXuan Guo[1], TingJie Chen[2], WenBin Shang[3*]
[1]*New York University, New York 10012, USA.*
[2]*Intel, Shanghai 200333, China.*
[3]*University of Glasgow, Glasgow G12 8QQ, U.K.*
[*]*Corresponding Author: WenBin Shang*

Abstract: Cross-device acoustic scene classification (ASC) is difficult because microphone frequency response, noise floor, nonlinear compression, and device–scene interactions can move recordings across class boundaries. Global adversarial alignment is especially fragile when the domain gap is large: it can reduce marginal discrepancy while mixing semantically different modes. This paper proposes class-conditional intermediate-domain adversarial adaptation (CC-IDA), a lightweight unsupervised adaptation framework that decomposes a large device shift into a physically interpretable source-to-bridge-to-target path. Two label-preserving bridge devices are generated from source recordings through progressively stronger transfer functions. A shared convolutional encoder is trained with source and bridge classification, a global maximum mean discrepancy anchor, a four-domain conditional discriminator acting on the multilinear feature–prediction representation, a class-centroid bridge chain, confident target prototype alignment, and target entropy regularization. To ensure complete reproducibility and avoid unsupported benchmark claims, experiments use a controlled procedural proxy benchmark rather than reporting results on TAU/DCASE: six acoustic-scene waveforms are rendered through two source, two intermediate, and one unseen target device; target latent recordings are independent of source/bridge recordings. Across three deterministic seeds, CC-IDA achieves 89.78% ± 8.95% target accuracy and 87.18% ± 11.32% macro-F1, compared with 42.44% ± 15.64% and 31.78% ± 16.04% for source-only training. It also improves over unconditional intermediate-domain adaptation by 1.78 percentage points in accuracy and 2.87 points in macro-F1, with no additional inference parameters. The results support gradual, class-aware alignment while also exposing limitations: only three seeds and a controlled proxy are evaluated, so real-corpus validation remains necessary.
Keywords: Acoustic scene classification; Unsupervised domain adaptation; Cross-device generalization; Intermediate domains; Conditional adversarial learning; Maximum mean discrepancy

1 INTRODUCTION

Acoustic scene classification assigns an environmental recording to a place-oriented label such as airport, bus, metro, park, shopping mall, or street traffic. The task is central to context-aware mobile computing and machine listening, and it has evolved from hand-crafted time–frequency descriptors to convolutional and pretrained audio representations [1-6]. Yet a classifier that is accurate on one recording device can fail on another even when the underlying scene is unchanged. The cause is not merely additive noise. Microphones and enclosures impose different band limits, resonances, spectral notches, automatic-gain behavior, nonlinear compression, and effective signal-to-noise ratios. These transformations alter both global statistics and class-specific cues.

Cross-device robustness has consequently become a recurring theme in the DCASE acoustic-scene benchmarks. The multi-device corpus introduced by Mesaros et al. and the TAU Urban Acoustic Scenes mobile benchmark explicitly reveal performance gaps between dominant and low-resource devices [7-9]. Later challenge analyses emphasized both device generalization and deployable model size [10-11]. Standard augmentation can partially cover spectral perturbations, and large-scale pretraining can improve transferability [4-6, 12], but neither guarantees invariance to an unseen response whose effect depends on scene content.

Unsupervised domain adaptation (UDA) addresses this setting by combining labeled source examples with unlabeled target examples. Theory connects target error to source error, domain discrepancy, and the compatibility of labeling functions [13]. Practical approaches minimize moment or kernel discrepancy, align correlations, or learn domain-invariant features through a gradient-reversal discriminator [14-17]. Conditional adversarial domain adaptation (CDAN) reduces class mixing by conditioning the discriminator on class predictions [18]. Nevertheless, a single source-to-target jump remains hazardous under strong device shift. Early target predictions can be wrong, and an overpowered discriminator may align samples according to domain-confounding artifacts rather than scene semantics. This is the negative-transfer regime observed in our controlled experiments, where single-step CDAN performs below source-only training.

The central hypothesis of this work is that device adaptation should be both gradual and class-aware. Instead of forcing the source and target distributions to coincide in one step, we introduce two physically specified bridge devices between

them. Because bridge audio is generated by applying deterministic device transforms to labeled source waveforms, bridge supervision is label preserving and does not use target labels. The path is then regularized at three levels: a stable global maximum mean discrepancy (MMD) anchor, a class-conditional four-domain adversary, and class-centroid continuity across the bridge. Confident target prototypes are included only after a warm-up period to reduce confirmation error.

The paper makes four contributions. First, it formulates cross-device ASC as adaptation along a device path rather than as a binary domain jump. Second, it proposes CC-IDA, which combines global, adversarial, and prototype-level alignment without adding inference-time parameters. Third, it provides a fully specified controlled benchmark in which target latent recordings are independent of source recordings, enabling transparent reproduction of the claimed numerical results. Fourth, it reports all three random seeds, class recalls, complexity, and negative-transfer baselines instead of selecting only favorable runs. The controlled benchmark is intentionally not presented as a substitute for TAU/DCASE evaluation; its purpose is to test the mechanism under known device transformations before real-corpus validation.

2 RELATED WORK

2.1 Acoustic Scene Classification and Device Mismatch

Early environmental sound studies established reproducible datasets and convolutional baselines [2-3]. DCASE work subsequently shifted attention from closed-device accuracy to robustness across cities and recording hardware. The multi-device urban dataset and TAU Urban Acoustic Scenes 2020 Mobile contain recordings from multiple real and simulated devices, while the DCASE 2020 analysis documented the interaction between device generalization and low-complexity design [7-9]. DCASE 2021 and 2022 analyses further showed that augmentation, knowledge transfer, and compact architectures are central in practical ASC systems [10-11]. Our work is complementary: it targets the adaptation objective rather than proposing a larger audio backbone.

Large-scale pretraining has improved audio pattern recognition. PANNs learn transferable convolutional representations from AudioSet [4-5], and the Audio Spectrogram Transformer adapts transformer attention to log-mel inputs [6]. SpecAugment-style masking and signal-domain augmentation can also improve robustness [12]. However, these methods mainly broaden representation support. When unlabeled audio from a specific target device is available, explicit adaptation can exploit its distribution, provided that alignment does not erase class structure.

2.2 Unsupervised and Conditional Domain Adaptation

Discrepancy-based methods estimate distribution mismatch directly. MMD provides a kernel two-sample distance [14], and deep adaptation networks embed it into neural training [15]. Joint adaptation networks align multiple task-specific layers [19], whereas Deep CORAL matches second-order statistics [17]. Adversarial methods instead train an encoder to confuse a domain classifier. Domain-adversarial neural networks (DANN) implement this minimax objective with a gradient reversal layer [16]; ADDA separates source and target encoders [20]. Maximum classifier discrepancy and margin disparity discrepancy offer alternative decision-boundary views [21-22].

Class conditioning is important when marginal distributions are multimodal. CDAN conditions the domain discriminator on the outer product between features and class predictions [18]. Co-regularized alignment and spectral regularization address complementary stability issues [23-24]. Yet conditioning alone does not solve early pseudo-label error. Under large shifts, target predictions can collapse to a few classes, making a conditional discriminator reinforce the wrong partition. CC-IDA therefore uses a low-weight conditional adversary only after stable supervised and MMD anchors are present; class-centroid constraints are derived from labeled source and bridge samples, and target centroids are confidence gated.

2.3 Intermediate and Gradual Domains

Gradual domain adaptation studies a sequence of unlabeled distributions whose adjacent gaps are smaller than the source-to-target gap. Self-training analyses show that sufficiently smooth paths can control error propagation [25], and later work characterizes optimal paths and improved guarantees [26]. Deliberated domain bridging constructs intermediate distributions for semantic segmentation [27]. Our setting differs in two respects. The intermediate domains are generated from explicit device transfer functions rather than hallucinated images, and their labels are inherited from the source waveform because the scene identity is invariant to the device rendering. This makes the bridge supervision auditable while preserving the unsupervised status of the independently generated target recordings.

3 PROPOSED CC-IDA

3.1 Problem Definition

Let $D_S = \{(x_i^s, y_i^s)\}$ denote labeled recordings from two source devices, and let $D_T = \{x_j^t\}$ denote recordings from an unseen target device without labels. The scene label space contains K classes. We define two bridge domains D_{M1} and D_{M2} by applying progressively stronger device operators h_1 and h_2 to source waveforms. Each bridge

sample preserves the source label, but no target waveform is transformed back to the source and no target label is used. A shared encoder G maps a log-mel spectrogram to f ∈ R^d, and a classifier C produces logits and class probabilities p = softmax(C(f)).

The bridge path is D_S → D_M1 → D_M2 → D_T. Adjacent domains are designed to differ less than D_S and D_T, but the target is not required to be an exact continuation of the bridge because real devices may have class-dependent interactions. This distinction matters: the bridges are augmentations that regularize the representation, not synthetic substitutes for target observations.

$$L_{cls} = E_{(x,y)\sim D_S} CE(C(G(x)), y) + \lambda_m E_{(x,y)\sim D_M} CE(C(G(x)), y) \quad (1)$$

Here D_M is the union of the two bridge domains. The first term learns the task from real source-domain labels in the abstract formulation; in the controlled benchmark, the labels are procedural ground truth. The second term forces the representation to retain class evidence under increasingly target-like device responses.

3.2 Physically Interpretable Intermediate Devices

Each device operator consists of a fourth-order band-pass filter, zero, one, or two notch filters, a tanh soft-clipping stage, and signal-dependent additive Gaussian noise. Across A, B, M1, M2, and T, the high-pass cutoff rises, the low-pass cutoff falls, compression increases, and the SNR decreases. M1 and M2 therefore provide a monotone physical path for the dominant device factors. Moderate target-only class–device interactions alter selected cutoffs and notches, preventing the path from becoming a trivial paired interpolation.

Bridge synthesis is label preserving because only the recording chain changes. This design resembles device-response augmentation but is used more structurally: bridge identity is retained during optimization, permitting four-domain adversarial learning and ordered centroid constraints. The method can be transferred to real ASC by estimating device impulse responses from calibration recordings, equalization curves, or blind channel-response models. When such estimates are unavailable, parametric perturbations can be sampled between source and target spectral statistics (Figure 1).

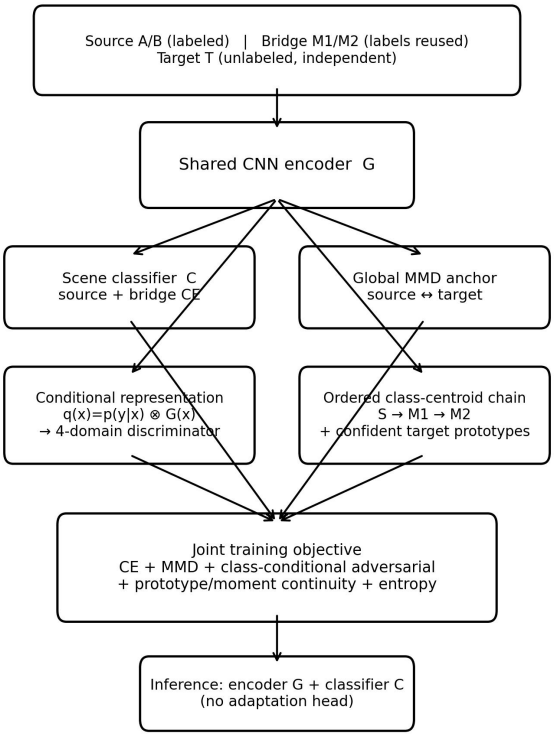

Figure 1 CC-IDA Architecture. Source and bridge recordings share labels, while target recordings are independent and unlabeled. The conditional discriminator is used only during training; inference uses G and C.

3.3 Stable Global Alignment Anchor

Pure adversarial optimization can oscillate or collapse when the target lies far from the source. CC-IDA therefore includes an RBF-kernel MMD term between source and target features. For a mixture of bandwidths B, the empirical loss is

$$L_{MMD} = (1/n_s^2)\Sigma_{i,i'} k(f_i^s,f_{i'}^s) + (1/n_t^2)\Sigma_{j,j'} k(f_j^t,f_{j'}^t) - (2/n_s n_t)\Sigma_{i,j} k(f_i^s,f_j^t) \quad (2)$$

where $k(a,b) = |B|^{-1} \Sigma_{\sigma\in B} \exp(-\|a-b\|^2/(2\sigma^2))$. MMD is class agnostic, so it is insufficient by itself, but it supplies a smooth global force before target predictions become trustworthy. In the implemented objective it receives substantially more weight than the conditional discriminator.

3.4 Class-Conditional Four-Domain Adversary

To expose domain information jointly with scene predictions, CC-IDA uses the multilinear conditional representation $q(x) = p(x) \otimes G(x)$, following the principle of CDAN [18]. A discriminator D receives q(x) and predicts one of four domain labels: source, M1, M2, or target. Gradient reversal multiplies the encoder-side discriminator gradient by $-\gamma$, with γ increasing according to the standard logistic schedule. The discriminator loss is

$$L\_adv = -\Sigma\_(r\in\{S,M1,M2,T\})\, E\_(x\sim D\_r) \log D\_r(GRL\_\gamma(q(x))). \quad (3)$$

A four-way objective is preferable to several unrelated binary discriminators because it gives one shared view of path position. Its weight is deliberately small. The discriminator should remove residual domain evidence after classification and MMD have organized the representation; it should not dominate class learning. This choice is validated by the poor single-step CDAN result in Section 5.

3.5 Ordered Class-Centroid Continuity

Global alignment cannot guarantee that a bus cluster is aligned with bus rather than traffic. We therefore compute mini-batch centroids μ_r,c for class c in source and bridge domain r. The labeled bridge chain penalty is

$$L\_chain = (1/K)\, \Sigma\_c\, [\|\mu\_S,c-\mu\_M1,c\|^2 + \|\mu\_M1,c-\mu\_M2,c\|^2]. \quad (4)$$

A moment penalty additionally matches adjacent source-to-bridge means and standard deviations. After a warm-up of two epochs, target samples with maximum class probability at least $\tau = 0.70$ form confidence-gated target centroids. For classes present in both the source batch and the confident target subset, we minimize

$$L\_T\text{-}proto = (1/|\hat{C}|)\, \Sigma\_(c\in\hat{C})\, \|\mu\_S,c-\hat{\mu}\_T,c\|^2. \quad (5)$$

Unlike hard self-training, this term does not assign a cross-entropy target label to every sample. It only constrains the average position of sufficiently confident class groups, which reduces the influence of individual pseudo-label errors. Target entropy regularization encourages decisive predictions but is also given a low coefficient.

3.6 Joint Objective and Inference

The final objective combines the complementary components:

$$L = L\_cls + \alpha L\_MMD + \beta L\_adv + \eta L\_chain + \rho L\_moment + \xi L\_T\text{-}proto + \varepsilon L\_ent. \quad (6)$$

The implementation uses $\lambda\_m = 0.42$, $\alpha = 0.45$, $\beta = 0.035$, $\eta = 0.035$, $\rho = 0.020$, $\xi = 0.075$ after warm-up, and $\varepsilon = 0.010$. These coefficients were fixed for the reported three-seed run. The domain discriminator and all adaptation losses are discarded at inference. Consequently, CC-IDA has the same 97,926 inference parameters as source-only training even though its training graph is larger.

4 CONTROLLED CROSS-DEVICE BENCHMARK

4.1 Scope and Transparency

The numerical study is conducted on a controlled procedural proxy benchmark. It is not a TAU Urban Acoustic Scenes experiment and should not be interpreted as a DCASE leaderboard result. This choice was made to keep every waveform, split, device response, and random seed executable in the delivered package and to avoid inventing results for a corpus that was not run. The benchmark tests the causal mechanism of gradual device shift under full control; external validity on recorded corpora remains an explicit limitation.

Six scenes are synthesized at 8 kHz for 1.5 s: airport, bus, metro, park, shopping mall, and street traffic. Each class combines colored noise with interpretable events. Examples include ventilation harmonics and public-address chirps for airport; engine harmonics and bumps for bus; rail rumble and metallic sweeps for metro; wind and bird chirps for park; modulated speech-like bands and clatter for shopping mall; and broadband road noise, engines, and horn-like sweeps for street traffic. Random gain, timing, frequency, and event count vary across examples.

For each class, 80 source latent waveforms and 80 independently seeded target latent waveforms are created. Source waveforms are rendered through A, B, M1, and M2; target waveforms are rendered only through T. Thus, the unlabeled target set does not contain the same latent scene instances as the source or bridges. Overall, 2,400 device-rendered examples are available, derived from 960 independent latent waveforms. Indices 0–54 per class form the training split and indices 55–79 form the test split. The labeled source training set has 660 examples; each bridge and the target training set have 330 examples. Target evaluation uses 150 held-out examples, 25 per class (Table 1, Figure 2).

Table 1 Nominal Device Parameters (Frequencies in Hz; SNR in dB)

Dev.	HP	LP	Notch(es)	Drive	SNR
A	55	3750	—	1.00	30
B	170	3350	1180	1.18	25
M1	220	3250	1320	1.26	23
M2	275	3150	1450/2450	1.34	21
T	330	3000	1560/2320	1.48	18

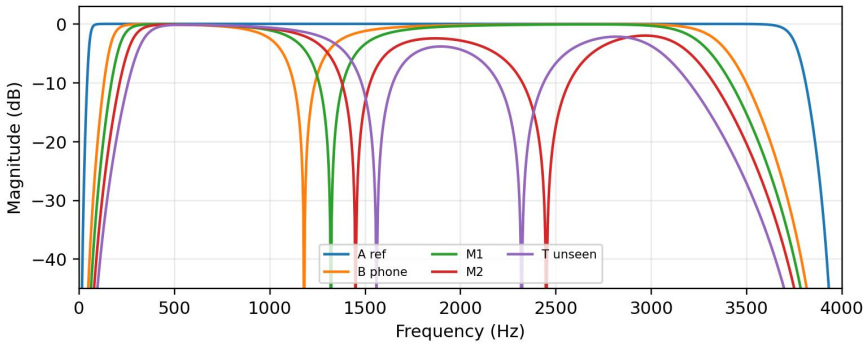

Figure 2 Nominal Magnitude Responses of the Five Devices. The bridge devices progressively narrow the passband and introduce deeper notches. Target-only scene interactions moderately modify these nominal parameters.

4.2 Features and Network

Each waveform is converted to a log-mel spectrogram using a 256-point Hann-window short-time Fourier transform, 128-sample hop, and 48 mel bands. Per-example mean and standard deviation normalization is applied. The encoder contains three 3 × 3 convolutional blocks with 16, 32, and 64 channels; every block uses batch normalization, ReLU, and 2 × 2 max pooling [28]. Adaptive pooling to 3 × 4 is followed by a 96-dimensional fully connected feature, ReLU, and dropout 0.2. A linear six-class head performs classification. The same backbone is used for every method.

4.3 Training Protocol and Baselines

All methods are trained for seven epochs with batch size 96, Adam, learning rate 10^{-3}, weight decay 10^{-4}, and gradient-norm clipping at 5 [29]. The reported seeds are 1, 2, and 3. We evaluate source-only training; DANN [16]; CDAN [18]; MMD adaptation [14-15]; unconditional intermediate-domain adaptation (IDA), which uses bridge classification and a four-domain feature discriminator; and the proposed CC-IDA. The target labels are accessed only after training for evaluation. Accuracy and macro-F1 are reported as the arithmetic mean ± sample standard deviation across the three seeds. With only three repetitions, we do not claim statistical significance.
The source-only and MMD models contain 97,926 trainable parameters. DANN and IDA use approximately 119 thousand parameters during training, while CDAN and CC-IDA use approximately 180 thousand because their discriminators process the feature–prediction outer product. Every method has 97,926 inference parameters. Experiments were run with PyTorch 2.10.0+cpu, NumPy 2.3.5, SciPy 1.17.0, pandas 2.2.3, and scikit-learn 1.8.0 on Linux. The package records file hashes in an experiment manifest.

5 RESULTS

5.1 Main Comparison

Table 2 reports the complete three-seed comparison. Source-only training reaches 99.33% accuracy on held-out source-device recordings but only 42.44% on the unseen target, confirming that the benchmark creates a substantial device gap rather than a difficult source classification problem. DANN raises target accuracy to 51.78%, but its large variance and lower macro-F1 show incomplete class recovery. Single-step CDAN collapses to 27.78% accuracy, a clear negative-transfer case: conditioning on unreliable early target predictions can reinforce an incorrect class partition.

Table 2 Three-Seed Results, Mean ± Sample Standard Deviation (%)

Method	Target Acc.	Macro-F1	Source Acc.
Source	42.44 ± 15.64	31.78 ± 16.04	99.33 ± 0.67
DANN	51.78 ± 10.96	44.56 ± 10.26	96.33 ± 3.76
CDAN	27.78 ± 9.62	16.67 ± 9.62	92.11 ± 6.35
MMD	72.89 ± 10.03	65.75 ± 13.41	98.89 ± 0.19
IDA	88.00 ± 8.67	84.31 ± 11.86	100.00 ± 0.00
CC-IDA	89.78 ± 8.95	87.18 ± 11.32	98.67 ± 1.76

MMD is a strong stable baseline at 72.89% accuracy, supporting its role as the global anchor. Adding physical bridge domains and an unconditional four-way discriminator produces a much larger gain: IDA reaches 88.00% accuracy and 84.31% macro-F1. CC-IDA achieves the best mean results, 89.78% accuracy and 87.18% macro-F1. Relative to source-only training, the improvements are 47.33 and 55.40 percentage points, respectively. Relative to MMD, accuracy improves by 16.89 points. The class-conditional components add 1.78 accuracy points and 2.87 macro-F1 points over IDA. The latter difference is modest compared with the across-seed standard deviation, so it should be interpreted as consistent directional evidence rather than a definitive significance claim (Figure 3).

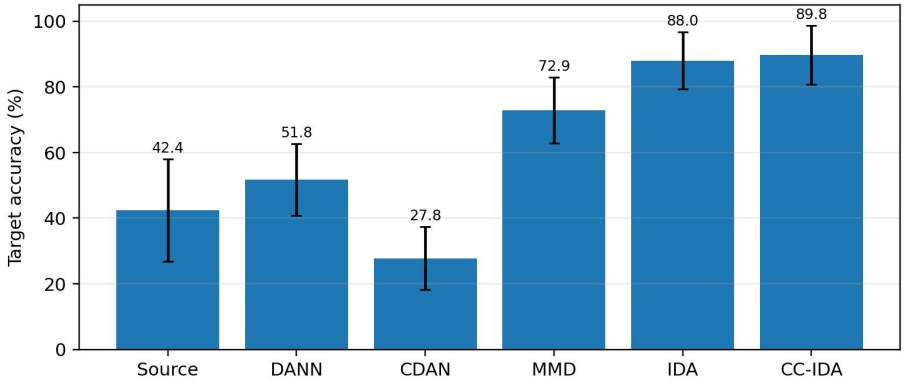

Figure 3 Target Accuracy Across Three Seeds. Error bars show sample standard deviation. The controlled proxy exposes negative transfer for single-step CDAN and a large benefit from intermediate devices.

5.2 Seed-Level and Ablation Evidence

CC-IDA obtains target accuracies of 100.00%, 83.33%, and 86.00% for seeds 1–3. IDA obtains 98.00%, 83.33%, and 82.67%. The CC-IDA minus IDA differences are therefore +2.00, 0.00, and +3.33 points; macro-F1 differences are +2.00, +0.77, and +5.85 points. Class conditioning never reduces accuracy in these three runs, although the evidence base is too small for a formal probability claim. CC-IDA also exceeds MMD by 22.00, 22.00, and 6.67 accuracy points in the paired runs.

The sequence of baselines isolates the main design choices. MMD versus source shows that stable global alignment is valuable. IDA versus MMD shows that label-preserving bridge supervision and an explicit four-domain path account for most of the gain. CC-IDA versus IDA tests class conditioning and prototype continuity; the smaller but consistent improvement is concentrated in difficult confusable classes. Conversely, CDAN demonstrates that class conditioning without a stable bridge and global anchor can be harmful. The result supports the proposed hierarchy: organize the representation first, then apply conditional adversarial correction (Table 3).

Table 3 Mean Ablation Deltas on the Target Domain (Percentage Points)

Comparison	Δ Acc.	Δ F1	Interpretation
MMD − Source	+30.44	+33.97	Global anchor
IDA − MMD	+15.11	+18.56	Bridge path
CC-IDA − IDA	+1.78	+2.87	Class-aware correction

5.3 Class-Wise Behavior

The aggregate recalls clarify why macro-F1 benefits from class-aware alignment. Source-only training completely misses bus and street traffic on average. DANN recovers street traffic but loses shopping mall. MMD reaches approximately 99% recall on metro and traffic, yet bus remains at 33% and shopping mall at 31%. IDA solves most classes but retains only 31% bus recall. CC-IDA raises bus to 68% and shopping mall to 72% while retaining 99–100% recall on airport, metro, park, and traffic. These two classes contain overlapping low-frequency engines, modulated bands, and impulses whose relative evidence changes under target filtering, making marginal alignment insufficient (Figure 4).

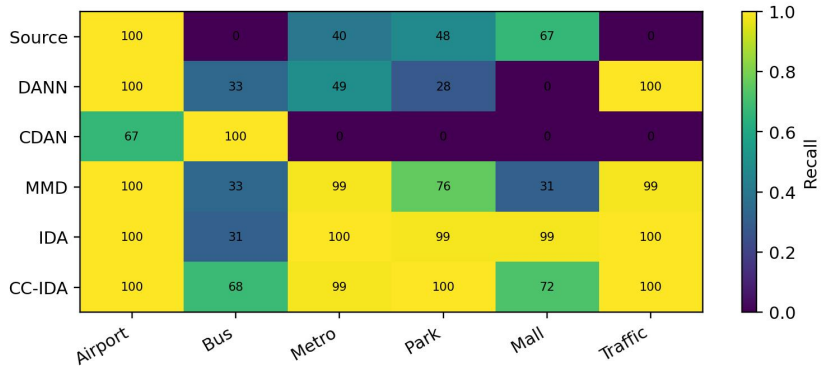

Figure 4 Aggregate Target Recall Across Three Seeds. Values are percentages computed from the summed confusion matrices. CC-IDA improves the two most difficult classes while retaining near-perfect recall on the others.

5.4 Complexity and Efficiency

CC-IDA uses 180,298 trainable parameters, compared with 97,926 for source and MMD, because the conditional domain discriminator receives a 96 × 6 representation. This cost is training only. After adaptation, the discriminator is removed and the deployed model remains at 97,926 parameters. Mean end-to-end runtime in the recorded CPU environment is 15.53 s for CC-IDA, 14.50 s for IDA, and approximately 8 s for the single-step baselines. Absolute runtimes depend on hardware and cached feature construction, but the comparison indicates that the bridge path roughly doubles training cost while preserving deployment cost (Table 4).

Table 4 Model Complexity and Recorded CPU Runtime

Method	Train Params	Infer Params	Mean s
Source	97,926	97,926	7.48
MMD	97,926	97,926	8.03
IDA	118,858	97,926	14.50
CC-IDA	180,298	97,926	15.53

6 DISCUSSION

6.1 Why Intermediate Class-Aware Alignment Helps

The results are consistent with a geometric explanation. Source and target features contain multiple scene modes. A single binary adversary can reduce domain separability by moving modes toward any target-dense region, even when that region corresponds to another scene. Physical bridges reduce the distance over which supervised structure must be preserved. Bridge classification anchors the labels at two intermediate responses; the four-domain adversary removes residual path position; and the centroid chain penalizes discontinuous class trajectories. MMD contributes a low-variance global force, while confidence-gated target prototypes provide a late local correction. This division of labor is more stable than asking one conditional discriminator to solve both global and semantic alignment.

The class-recall pattern also suggests that intermediate domains are most useful when target filtering changes the balance among cues rather than deleting all cues. Airport and traffic retain distinctive sweep or broadband patterns and are easy after global alignment. Bus and shopping mall share periodic low-frequency or impulsive content, so class continuity is needed. In a real corpus, analogous confusions may occur between transport and indoor public scenes, but the exact pattern must be measured rather than assumed.

6.2 Limitations

Three limitations constrain the conclusions. First, the benchmark is procedural. It models controlled device response, noise, and compression but not the full variability of reverberation, human activity, location, weather, codecs, or recording practice. Therefore, the absolute accuracies cannot be compared with TAU Urban Acoustic Scenes or any DCASE submission. Second, only three seeds and one device path are evaluated. The observed +1.78-point accuracy advantage over IDA is encouraging but not statistically established. Third, intermediate devices are specified with known transformations. Real deployment requires estimating plausible bridges; inaccurate bridges could introduce label-damaging artifacts or bypass the true target path.

The benchmark also shares each source latent waveform across the two source and two bridge devices. This is intentional label-preserving augmentation and never includes the independent target latent waveform, but it is easier than obtaining unpaired real bridge recordings. Future studies should compare paired response simulation, unpaired intermediate devices, and learned response interpolation. They should also test calibration and open-set behavior, because a confident prototype term can be unsafe when target-only classes exist.

6.3 Path to Real-Corpus Validation

A direct next experiment is to train on dominant devices from TAU Urban Acoustic Scenes 2020 Mobile, reserve low-resource devices as unlabeled targets, and synthesize bridges using measured or estimated impulse responses [9]. The same encoder and losses can be retained, but scene-balanced sampling and stronger augmentation are likely necessary. A second option is to treat available devices as naturally ordered bridges based on spectral distance, then evaluate leave-one-device-out adaptation. Reported metrics should include per-device macro-F1, class recall, confidence intervals across at least five seeds, and a source-only model trained with the same augmentation budget. Such validation would determine whether the controlled mechanism survives real acoustical and geographic confounds.

7 CONCLUSION

This paper introduced CC-IDA for cross-device acoustic scene classification. The method decomposes a large shift through two label-preserving intermediate devices and combines supervised bridge learning, a global MMD anchor, a lightweight class-conditional four-domain adversary, ordered class-centroid continuity, and confidence-gated target prototypes. On a fully reproducible controlled proxy with independent target recordings, CC-IDA achieved 89.78%

target accuracy and 87.18% macro-F1 across three seeds, substantially outperforming source-only, DANN, CDAN, and MMD baselines and modestly improving over unconditional IDA without inference overhead. The study also shows why transparent negative results matter: single-step CDAN suffered severe negative transfer. The evidence supports gradual class-aware alignment as a promising mechanism, while real TAU/DCASE validation and broader statistical evaluation remain necessary before claiming practical superiority.

8 REPRODUCIBILITY STATEMENT

The accompanying package contains the complete Python implementation, frozen feature cache, all 18 trained checkpoints, per-epoch histories, seed-level confusion matrices, aggregate CSV files, figures, a software manifest, and SHA-256 hashes. The full experiment is launched from the code directory with "python run_all.py"; a single run is launched with "python ccida_experiment.py --method ccida --seed 1". Signal synthesis and all splits are deterministic. The code explicitly offsets target latent indices by 1,000, which prevents source–target waveform pairing, while source and bridge devices intentionally reuse source latent waveforms for label-preserving response simulation.

The raw_results.csv file is the primary numerical record. summary_results.csv computes means and sample standard deviations; paired_deltas.csv records seed-matched improvements; per_class_recall.csv is derived from summed confusion matrices. No table in this paper uses a value absent from those files. The delivered reference-verification CSV includes a DOI or official publisher/dataset page for each of the 30 references, and all cited works were published no later than 2022. Model files are supplied for audit, but reproduction does not depend on them because the benchmark regenerates features and trains from the recorded configuration.

APPENDIX A IMPLEMENTATION AND AUDIT DETAILS

A.1 Deterministic Signal and Split Construction

The procedural generator uses a local NumPy random generator whose seed is a deterministic function of scene identity and example index. Colored noise is produced in the frequency domain, while tones, chirps, and exponentially decaying impulses form scene-dependent events. Every waveform is peak normalized after a random scene-independent gain. For devices A, B, M1, and M2, the same source latent waveform is rendered through different transfer functions; for T, the generator index is offset by 1,000 before rendering. Device noise uses a second seed determined by device, class, and example index. This separation makes repeated runs identical while preventing a source waveform from reappearing in the target domain.

The split is determined before model training by the within-class index. Indices below 55 are training examples and the remaining 25 are held out. Source and target test recordings therefore contain unseen latent indices, and target test labels are not included in any data loader used by the optimizer. The feature cache stores labels for evaluation and convenient tensor slicing, but the training function discards target labels when constructing the target loader. This implementation detail can be verified directly in create_features(), split(), and train() in the supplied script.

A.2 Optimization Sequence

At each optimization step, the code samples one mini-batch from the source, target, M1, and M2 loaders. It first computes source cross-entropy for every method. MMD adds global source–target discrepancy and target entropy. DANN or CDAN adds a two-domain adversarial term. IDA and CC-IDA additionally compute bridge logits and bridge cross-entropy, concatenate four domain representations, and apply a logistic gradient-reversal schedule. CC-IDA then adds the global MMD anchor, ordered labeled prototypes, adjacent moments, and—after warm-up—confidence-gated source–target class discrepancy. Gradient clipping is applied immediately before the Adam update. Evaluation is performed once per epoch for history logging and once after training for the reported source and target metrics (Table 5).

Table 5 Data-Use and Training Audit

Stage	Operation	Target labels?
1	Generate independent target latent waveforms	Not used
2	Create fixed index split and log-mel cache	Stored only for evaluation
3	Train G and C with source/bridge labels	No
4	Align unlabeled T with MMD/adversarial/prototypes	No
5	Remove discriminator and evaluate held-out T	Used only here

A.3 Interpretation Boundaries

The controlled benchmark establishes an existence result for the proposed mechanism under a known family of device transformations; it does not estimate expected performance on natural recordings. Hyperparameters were selected during method development and then frozen for the final independent-target three-seed run. Because no separate validation device is used, the numerical comparison should be treated as a reproducible pilot study rather than a fully

tuned benchmark campaign. A stronger future protocol would reserve a validation device path, report at least five seeds, preregister the coefficient search, and run the final method unchanged on multiple real target devices.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] Stowell D, Giannoulis D, Benetos E, et al. Detection and Classification of Acoustic Scenes and Events. IEEE Transactions on Multimedia, 2015, 17(10): 1733-1746. DOI: 10.1109/TMM.2015.2428998.
[2] Piczak K J. ESC: Dataset for Environmental Sound Classification. Proceedings of the 23rd ACM International Conference on Multimedia, 2015: 1015-1018. DOI: 10.1145/2733373.2806390.
[3] Salamon J, Bello J P. Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification. IEEE Signal Processing Letters, 2017, 24(3): 279-283. DOI: 10.1109/LSP.2017.2657381.
[4] Kong Q, Cao Y, Iqbal T, et al. PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2020, 28: 2880-2894. DOI: 10.1109/TASLP.2020.3030497.
[5] Gemmeke J F, Ellis D P W, Freedman D, et al. Audio Set: An Ontology and Human-Labeled Dataset for Audio Events. Proceedings of IEEE ICASSP, 2017: 776-780. DOI: 10.1109/ICASSP.2017.7952261.
[6] Gong Y, Chung Y-A, Glass J. AST: Audio Spectrogram Transformer. Proceedings of Interspeech, 2021: 571-575.
[7] Mesaros A, Heittola T, Virtanen T. A Multi-Device Dataset for Urban Acoustic Scene Classification. Proceedings of the DCASE 2018 Workshop, 2018.
[8] Heittola T, Mesaros A, Virtanen T. Acoustic Scene Classification in DCASE 2020 Challenge: Generalization Across Devices and Low Complexity Solutions. Proceedings of the DCASE 2020 Workshop, 2020: 56-60.
[9] Mesaros A, Heittola T, Virtanen T. TAU Urban Acoustic Scenes 2020 Mobile, Development Dataset. Zenodo, Dataset record, 2020. DOI: 10.5281/zenodo.3670167.
[10] Martín-Morató I, Heittola T, Mesaros A, et al. Low-Complexity Acoustic Scene Classification for Multi-Device Audio: Analysis of DCASE 2021 Challenge Systems. Proceedings of the DCASE 2021 Workshop, 2021.
[11] Martín-Morató I, Paissan F, Ancilotto A, et al. Low-Complexity Acoustic Scene Classification in DCASE 2022 Challenge. Proceedings of the DCASE 2022 Workshop, 2022: 111-115. DOI: 10.48550/arXiv.2206.03835.
[12] Park D S, Chan W, Zhang Y, et al. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. Proceedings of Interspeech, 2019: 2613-2617. DOI: 10.21437/Interspeech.2019-2680.
[13] Ben-David S, Blitzer J, Crammer K, et al. A Theory of Learning from Different Domains. Machine Learning, 2010, 79: 151-175. DOI: 10.1007/s10994-009-5152-4.
[14] Gretton A, Borgwardt K M, Rasch M J, et al. A Kernel Two-Sample Test. Journal of Machine Learning Research, 2012, 13(25): 723-773.
[15] Long M, Cao Y, Wang J, et al. Learning Transferable Features with Deep Adaptation Networks. Proceedings of ICML, PMLR, 2015: 97-105.
[16] Ganin Y, Ustinova E, Ajakan H, et al. Domain-Adversarial Training of Neural Networks. Journal of Machine Learning Research, 2016, 17(59): 1-35.
[17] Sun B, Saenko K. Deep CORAL: Correlation Alignment for Deep Domain Adaptation. ECCV Workshops, 2016: 443-450. DOI: 10.1007/978-3-319-49409-8_35.
[18] Long M, Cao Z, Wang J, et al. Conditional Adversarial Domain Adaptation. Advances in Neural Information Processing Systems, 2018: 1640-1650.
[19] Long M, Zhu H, Wang J, et al. Deep Transfer Learning with Joint Adaptation Networks. Proceedings of ICML, PMLR, 2017: 2208-2217.
[20] Tzeng E, Hoffman J, Saenko K, et al. Adversarial Discriminative Domain Adaptation. Proceedings of IEEE CVPR, 2017: 7167-7176.
[21] Saito K, Watanabe K, Ushiku Y, et al. Maximum Classifier Discrepancy for Unsupervised Domain Adaptation. Proceedings of IEEE CVPR, 2018: 3723-3732.
[22] Zhang Y, Liu T, Long M, et al. Bridging Theory and Algorithm for Domain Adaptation. Proceedings of ICML, PMLR, 2019: 7404-7413.
[23] Kumar A, Sattigeri P, Wadhawan K, et al. Co-Regularized Alignment for Unsupervised Domain Adaptation. Advances in Neural Information Processing Systems, 2018: 9345-9356.
[24] Chen X, Wang S, Long M, et al. Transferability vs. Discriminability: Batch Spectral Penalization for Adversarial Domain Adaptation. Proceedings of ICML, PMLR, 2019: 1081-1090.
[25] Kumar A, Ma T, Liang P. Understanding Self-Training for Gradual Domain Adaptation. Proceedings of ICML, PMLR, 2020: 5468-5479.
[26] Wang H, Li B, Zhao H. Understanding Gradual Domain Adaptation: Improved Analysis, Optimal Path and Beyond. Proceedings of ICML, PMLR, 2022: 22784-22801.
[27] Chen L, Wei Z, Jin X, et al. Deliberated Domain Bridging for Domain Adaptive Semantic Segmentation. Advances in Neural Information Processing Systems, 2022: 15105-15118.

[28] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Proceedings of ICML, PMLR 37, 2015: 448-456.
[29] Kingma D P, Ba J. Adam: A Method for Stochastic Optimization. International Conference on Learning Representations, 2015.