Perch-Distilled ProtoSSM with Rigorous Post-Processing: A Case Study on BirdCLEF+ 2026

Yifei Deng[1,*,†], Shane Deng[2,*,†]

[1]Guangdong University of Foreign Studies, Guangzhou, China
[2]Independent Researcher, Guangzhou, China

Abstract

This paper presents a highly efficient bioacoustic classification pipeline for the BirdCLEF+ 2026 challenge, achieving a Private Leaderboard ROC AUC score of 0.944 and a Silver Medal through a heterogeneous ensemble with a community-sourced Sound Event Detection (SED) model. To overcome the strict 90-minute CPU-only inference constraint, we propose a novel paradigm coupling the spatial priors of the pre-trained Perch V2 foundation model—optimized via an open-source ONNX runtime—with an extremely lightweight state space sequence model (ProtoSSM), compressing the entire pipeline's execution to under 25 minutes. The dominance of the single ProtoSSM architecture (Private LB ROC AUC 0.935) stems from multi-dimensional synergies: cost-free "feature-level Mixup" for long-tail regularization, a two-stage ResidualSSM for logit-space correction, and probability-preserving post-processing techniques guided by biological priors. Furthermore, a post-competition diagnostic system using forced probability truncation revealed that ProtoSSM's superiority strictly depends on continuous temporal context; while it excels on 1-minute environmental recordings, it degrades into a shallow feature wrapper without temporal fuel. Ultimately, this work offers a rigorous engineering template and deep physical insights for deploying low-resource bioacoustic sequence models in real-world edge computing scenarios. Supporting code for this paper can be found at https://github.com/DYF7812/BirdCLEF-2026.

Keywords

Bioacoustic Classification, Prototypical State Space Model (ProtoSSM), Feature Distillation, Probabilistic Post-processing

1. Introduction

The BirdCLEF+ 2026 challenge focuses on the Pantanal wetlands of Brazil—a vast natural reserve under severe ecological threat, urgently requiring the establishment of scalable biodiversity monitoring networks. Participants are tasked with accurately identifying 234 target taxa (spanning avian, amphibian, mammalian, reptilian, and insect vocalizations) from continuous passive acoustic monitoring (PAM) environmental audio [1, 2]. Although deep learning has revolutionized the field of bioacoustics, model deployment on real-world edge devices is often constrained by severe computational bottlenecks. To accurately simulate this practical dilemma, this Kaggle competition enforced a stringent 90-minute, CPU-only inference limit. The official evaluation metric for the competition is the unweighted macro ROC AUC across all target classes; therefore, all Public and Private Leaderboard (LB) scores reported throughout this paper refer to this ROC AUC metric unless otherwise specified.

The core pain point of this task is the insurmountable domain shift between the training and test sets. This year's official dataset consists of over 30,000 isolated short focal recordings and a severely scarce set of 66 annotated 1-minute environmental audio clips. Looking at the champion and runner-up solutions of past BirdCLEF competitions, the mainstream breakthrough paradigm typically involves stacking massive ensembles of Convolutional Neural Networks (CNNs) and relying on iterative brute-force pseudo-labeling on unannotated environmental audio to forcefully fit the domain distribution [3, 4]. This "brute-force aesthetics" based on ensembling and massive pseudo-labels has shone brilliantly in past editions and continues to dominate the top tier of this year's leaderboard [5, 6].

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany
*Corresponding author.
†These authors contributed equally.
781276719@qq.com (Y. Deng); dengsx@gmail.com (S. Deng)

However, the emergence of those 66 highly precious 1-minute continuous environmental recordings opened an unprecedented avenue for exploring lightweight sequence modeling. This paper breaks away from the traditional CNN ensembling paradigm, opting instead to explore the deep synergy between universal acoustic foundation models (Perch V2) and cutting-edge State Space Models (SSMs) for efficient sequence processing. Specifically, we first utilized an open-source, highly optimized ONNX version of Google's Perch V2 [7] to extract high-dimensional spatial priors under strict CPU constraints. Subsequently, we utilized these dense embeddings as input features, distilling them into a highly customized Prototypical State Space Model (ProtoSSM) and an auxiliary residual network (ResidualSSM). This lightweight architecture makes a breakthrough by perfectly mapping isolated static frame features onto the coherent acoustic rhythms and cadences of physical reality, bridging the massive gap between short focal recordings and long environmental audio at a fundamental physical level.

To further squeeze out the prediction confidence of the single architecture, we constructed an extremely robust post-processing defense line in the probability space. This calibration suite, based on mathematical and physical priors, encompasses per-taxa temperature scaling, taxonomy smoothing, temporal smoothing, and Top-1 Power file-level scaling. At the end of this paper, we present a highly rigorous and straightforward post-competition black-box diagnostic report. This report aims to systematically deconstruct the true sources of gains from foundation model feature distillation, precisely delineate the capability boundaries of sequence architectures in bioacoustics, and bluntly dissect their inherent performance limitations when facing extremely low-resource classes (lacking foundation priors and temporal support).

2. Related Work

The paradigm of bioacoustic classification is rapidly shifting toward foundation models pre-trained on massive, multi-taxa datasets. Perch 2.0, released by Google DeepMind in early 2026, represents the latest milestone in this domain [8]. Based on an EfficientNet-B3 backbone, the model underwent strong supervised training on a massive dataset containing over 1.5 million annotated recordings across 14,795 classes, spanning birds, amphibians, mammals, and insects. By introducing prototype-learning self-distillation and a novel source-prediction auxiliary objective, Perch v2 outputs highly robust 1536-dimensional embeddings and demonstrates dominating generalization capabilities in few-shot transfer and retrieval tasks. However, constrained by its architectural receptive field, Perch can only operate isolatedly on independent 5-second short audio windows. This means that while its acoustic feature extraction on single frames is nearly flawless, it entirely loses the cross-window global perspective, lacking the mechanism to explicitly model the inherent temporal continuity in real-world long recordings (such as complex rhythmic calling and inter-species response patterns within a 1-minute sequence).

To bridge the representational gap between static single-frame features and continuous environmental audio streams, robust sequence modeling is indispensable. Although Transformer architectures have long dominated the field of sequence modeling, the computational and memory complexity of their self-attention mechanism, which grows quadratically with sequence length, makes them impractical for processing long audio in the wild and facing edge computing bottlenecks. In recent years, Selective State Space Models (Selective SSMs)—particularly the Mamba architecture [9]—have emerged as a highly efficient and disruptive alternative. By introducing input-dependent parameters and hardware-aware selective state update mechanisms, Mamba successfully achieves precise capture of global receptive fields for long sequences while maintaining linear computation time and extremely low memory footprint.

Over the past year or two, SSM architectures have been rapidly ported to continuous audio representation learning. For example, AcousticSSM [10] proposed by Liang et al. innovatively integrated a convolutional spatial feature extractor with an SSM backbone, demonstrating the remarkable capability of SSMs in extracting deep temporal long-range dependencies in acoustic few-shot classification tasks.

Concurrently, cutting-edge research in ecoacoustics introduced the BioMamba model [11], which was exhaustively validated on the BEANS benchmark covering a wide range of bioacoustic tasks. Its core conclusion firmly established that when processing ultra-long environmental audio sequences, Mamba-based architectures can achieve equivalent or superior recognition and detection performance with far lower VRAM and computational overhead than traditional Transformer paradigms (such as AVES).

When tackling the challenges of long-tail distributions and few-shot classification, the design of the metric space is just as crucial as optimizing the spatio-temporal feature extraction capability of the backbone network. Among these, the classic Prototypical Networks proposed by Snell et al. [12] provide an extremely elegant solution. This theory advocates that by explicitly learning and maintaining a representative vector center (i.e., a "prototype") for each class in the latent space, the model can simply perform classification based on the distance metric (such as Euclidean distance or cosine similarity) between the feature embeddings and each prototype, without relying on complex fully connected layers that are prone to overfitting.

This metric learning-based paradigm not only greatly reduces the risk of network overfitting under extremely low-resource data but also endows the classification latent space with high physical and geometric interpretability. In the recent evolution of ecoacoustic models, coupling the robust spatial representations extracted by foundation models with the metric classification of prototypical networks has increasingly proven to be an effective approach to enhancing model generalization in noisy field audio. This solid prior theory also laid a critical algorithmic foundation for the subsequent construction of our innovative architecture, which "anchors SSM temporal inference with a prototype-based dynamic metric space."

3. Methodology

3.1. Overall Pipeline

Our overall architecture for this competition starts with Perch acoustic feature extraction and spans two-stage model training and complex post-processing fusion strategies. First, ProtoSSM Stage 1 completes basic training using pre-cached Perch features extracted from 66 Soundscapes audio files. Subsequently, in Stage 2, pseudo-labeled data is proportionally mixed in for augmentation, aiming to enhance the model's recognition robustness on classes lacking environmental audio training data. The prediction outputs of these two stages are robustly fused via RMS alignment at the logit level, followed by the introduction of ResidualSSM to perform fine-grained residual correction on the fused logits. Before entering the multi-model ensembling stage, the probability distribution of the backbone network undergoes a series of strict post-processing calibrations, including per-taxa temperature scaling, taxonomy smoothing, temporal smoothing, and Top-1 Power file-level scaling. Concurrently, we independently loaded the community open-source SED ONNX model [13] for parallel inference, and utilized dimensionality-reduced Perch embeddings alongside short-audio cached features to train a CatBoost (CAB) tree model to provide auxiliary diagnostics. The prediction results from the backbone ProtoSSM, SED, and CAB are ultimately mapped to the Rank space (percentile ranking), achieving late-stage heterogeneous fusion through a weight matrix dynamically allocated based on a class grouping strategy. Finally, the entire pipeline is concatenated with a global filtering and enhancement mechanism, encompassing noise and SED spike suppression, temporal window smoothing based on ProtoSSM confidence, synonymous alignment of strongly correlated species (Mirror Pairs), and row-wise contextual outlier correction and enhancement (Row-wise Suppression / Enhancement), thereby yielding the final submission results. This complete pipeline achieved a Public LB score of 0.950 (ranking 1200+), while securing an outstanding 0.944 on the final Private LB, leaping to 115th place [14].

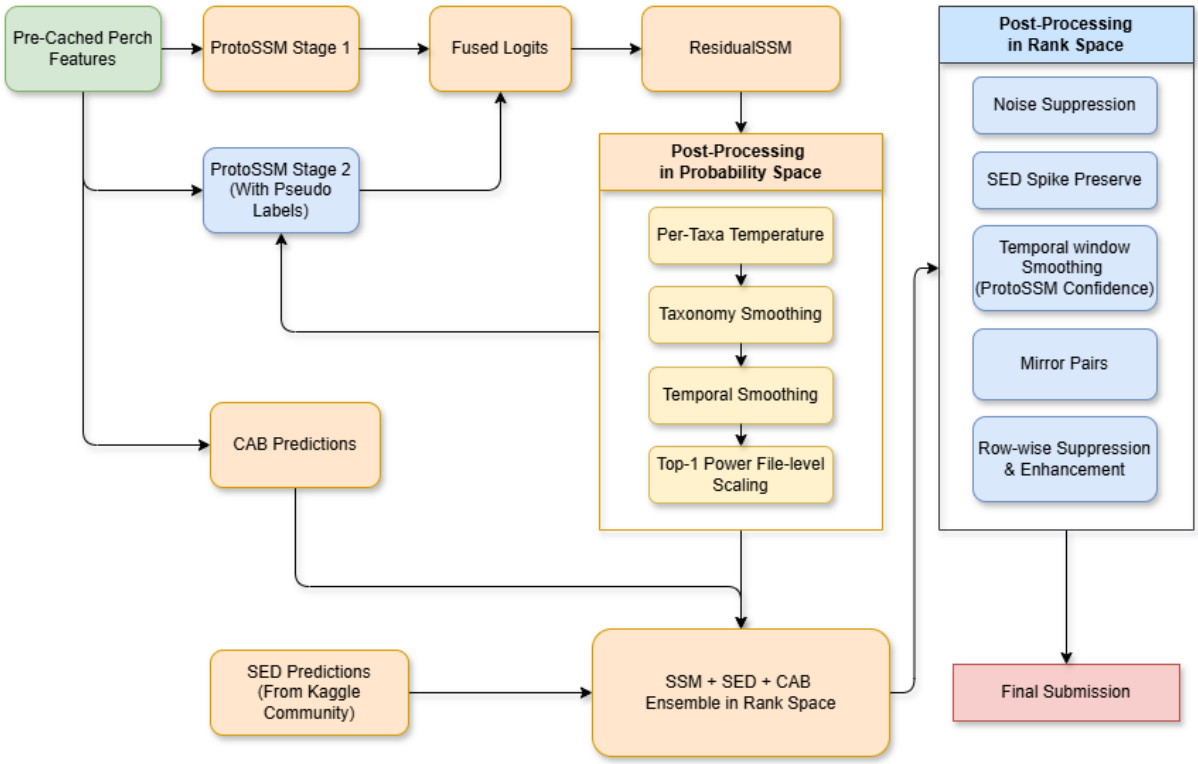

Figure 1: Overall architecture pipeline of our BirdCLEF+ 2026 submission, illustrating the flow from Perch feature extraction through ProtoSSM two-stage training, ResidualSSM correction, probability post-processing, and heterogeneous late fusion.

3.2. Data Preprocessing

Before entering ProtoSSM architecture training, standardized feature extraction and pre-caching must be performed on the raw audio data. This decouples the time-consuming acoustic processing from the subsequent sequence model training. The core steps of this preprocessing pipeline include:

1. **Label File Deduplication and Aggregation**: The input label files, such as `train_soundscapes_labels.csv`, are first cleaned by performing deduplication at both the paragraph (Row-level) and class label (Token-level) levels. For multiple observations that may occur within the same 5-second time window (Segment), their `primary_labels` are aggregated via union to ensure each window has a unique set of ground truth labels.
2. **ONNX-Accelerated Perch Feature Extraction**: The ONNX-optimized Perch v2 model is employed to infer on audio slices. For each 5-second window, it outputs a 1536-dimensional dense embedding vector and multi-class logits.
3. **Key Taxonomic Family Feature Aggregation**: In addition to extracting basic class logits, the pipeline specifically aggregates and calculates the mean and max values of the corresponding logits for 5 specific families: Amphibia, Aves, Insecta, Mammalia, and Reptilia. These are cached as high-order prior features.
4. **Metadata Extraction and Row ID Alignment**: For Soundscapes files, contextual metadata such as recording location (Site), date (Date), and hour (Hour) are parsed directly from the audio filenames using regex matching.
5. **Building the Local Feature Cache**: The extracted information above is uniformly aligned along the time axis based on a unique `row_id` and saved into independent caches by category. Metadata is saved in `.parquet` files, while core feature data is serialized into `.npz` files (compressed via `np.savez_compressed`). Within the `.npz` format, features are strictly partitioned into multiple

independent key arrays sharing the first dimension N (number of samples arranged by time-axis windows):

- `embeddings`: Extracted dense acoustic features ($N \times 1536$).
- `scores`: Predicted logits for basic classes ($N \times 234$).
- `labels`: Multi-hot encoded ground truth classification labels, i.e., the Y matrix ($N \times 234$).
- `masks`: A label mask matrix indicating whether the current 5-second audio segment is valid ($N \times 1$). For the 66 files with ground truth labels, unannotated windows are masked as 0, while annotated windows are set to 1.
- `family_max_te` and `family_mean_te`: The previously calculated family-level statistical features are also saved in the cache.

All matrices share the same Row Index to ensure absolute row-level alignment with the metadata table.

3.3. ProtoSSM Architecture

3.3.1. Baseline Version

Our core sequence modeling network's Baseline Version borrows designs from an early ProtoSSM V2 version in the public notebook: pantanal-distill-birdclef2026 [15]. It employs a prototypical state space model that fuses metadata with a Bidirectional Selective State Space (Bidir-SSM). This architecture aims to combine the static features extracted by the foundation model (Perch) with contextual metadata to accomplish long-range dependency modeling along the temporal axis. Its specific architecture and forward propagation flow are as follows:

1. **Input Projection & Positional Encoding**
   The input features are the frame-level embedding vectors extracted by Perch v2 (shape $[B, T, D_{\text{input}}]$, where $T = 12$ is the sequence length of audio slices). The embeddings first pass through a projection layer consisting of Linear Projection, LayerNorm, GELU activation, and Dropout, mapping them to a unified hidden dimension d_{model}. To preserve absolute temporal structure, the model introduces learnable positional encoding parameters and adds them directly to the projected features.
2. **Metadata Fusion**
   Considering that species vocalization behavior is strongly correlated with geographic location and time, the baseline architecture integrates the contextual information of Site and Hour. Categorical metadata are mapped to low-dimensional embeddings and transformed to dimension d_{model} via a linear projection layer, finally being fused into the backbone sequence features additively:
   $$h = h + \text{meta\_proj}([\text{Emb}_{\text{site}}, \text{Emb}_{\text{hour}}]) \tag{1}$$
3. **Bidirectional Selective SSM Layers**
   The core of the sequence modeling consists of multiple stacked layers of Bidirectional Selective State Space modules. Each layer contains independent forward and backward `SelectiveSSM` units. The feature sequence passes through the forward SSM and the backward SSM (after reversing the time steps, and then reversing the output back to normal order) in parallel, achieving full extraction of bidirectional context. The bidirectional outputs are concatenated and restored to the d_{model} dimension via a linear layer, and finally regularized using a Post-Norm residual structure.
4. **Prototypical Metric Learning**
   Compared to traditional linear classification heads, ProtoSSM uses prototype-based metric learning. The model maintains a learnable prototype vector (shape $[N_{\text{classes}}, d_{\text{model}}]$) for each species class. It calculates the cosine similarity on a unit hypersphere between the sequence feature h and each species prototype, calibrating it using a learnable scaling temperature parameter (ensured positive by Softplus activation) and a class bias:
   $$\text{sim} = \text{sim}_{\cos}(h, \text{Prototypes}) \times \text{Softplus}(\text{temp}) + \text{bias} \tag{2}$$

5. **Dynamic Teacher Fusion**
   To address the limitations of foundation model features and introduce prior knowledge, the model performs dynamic feature-level fusion on the calculated prototype similarity scores and the original predictions of the Perch teacher model (Perch Logits). A learnable parameter α (constrained between 0 and 1 via Sigmoid activation) is introduced to determine the dependency weight of each class on prototype predictions versus teacher predictions:

$$\text{species\_logits} = \alpha \times \text{sim} + (1 - \alpha) \times \text{perch\_logits} \tag{3}$$

   For classes that have no mapping relationship (Unmapped) in the Perch model, α can be forced to 1.0, relying entirely on the prototype path for prediction.
6. **Loss Function Design**
   To achieve end-to-end supervised learning and knowledge distillation, the total training loss function of the baseline model is composed of a weighted sum of the classification loss and the teacher prediction distillation loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{BCE}} + \lambda_{\text{dist}}\mathcal{L}_{\text{dist}} \tag{4}$$

   BCE Loss with Focal Modulation: A binary cross-entropy loss is calculated for each time step. To balance the ratio of positive to negative samples, a class-level positive sample weight bias is introduced. Simultaneously, a Focal Loss modulation coefficient is used:

$$\mathcal{L}_{\text{BCE}} = \frac{1}{B \cdot T \cdot C} \sum_{b,t,c} (1 - p_t)^{\gamma} \cdot \text{BCE}(y_{b,t,c}, \hat{y}_{b,t,c}) \tag{5}$$

   where p_t is the closeness of the predicted probability to the ground truth label, and γ is the Focal modulation factor.
   Knowledge Distillation Loss: Knowledge transfer is achieved by calculating the Mean Squared Error (MSE) between the model outputs and the predicted logits of the Perch v2 foundation model. This loss is only applied to species classes that have a corresponding mapping in Perch ($w_c^{\text{distill}} = 1$):

$$\mathcal{L}_{\text{dist}} = \frac{1}{B \cdot T \cdot C} \sum_{b,t,c} (\hat{z}_{b,t,c} - z_{b,t,c}^{\text{Perch}})^2 \cdot w_c^{\text{distill}} \tag{6}$$

7. **Cross-Validation Strategy**
   The baseline version employs basic GroupKFold for a 5-fold dataset split. During splitting, a Group Key (combination rule: source + file_id + site) is constructed based on the extracted audio filenames, ensuring that clips from the same recording device and location are strictly isolated across folds to prevent data leakage.

3.3.2. Enhanced Version

Building upon the baseline, the Enhanced Version expands the network's capabilities through a series of configuration parameters (independently controllable in CFG), facilitating comparison and ablation experiments between the two versions:

1. **Temporal Masking Support**: Allows the model to input audio data lacking annotations for the full 12 windows, maximizing the utilization of all annotations. A time step mask $m_{b,t} \in \{0, 1\}$ is introduced during loss calculation, ensuring BCE and distillation losses are only computed on windows containing valid annotations:

$$\mathcal{L}_{\text{BCE\_masked}} = \frac{\sum_{b,t,c} (1 - p_t)^{\gamma} \cdot \text{BCE}(y_{b,t,c}, \hat{y}_{b,t,c}) \cdot m_{b,t}}{\sum_{b,t} m_{b,t} + \epsilon} \tag{7}$$

2. **Pseudo-Label Confidence Penalty**: To smooth out the noise introduced by pseudo-labels in Stage 2, the enhanced version introduces a class confidence penalty weight w_c^{pseudo} for pseudo-labeled samples:

$$\mathcal{L}_{\text{BCE\_pseudo}} = \frac{\sum_{b,t,c}(1-p_t)^{\gamma} \cdot \text{BCE}(y_{b,t,c}, \hat{y}_{b,t,c}) \cdot m_{b,t} \cdot w_c^{\text{pseudo}}}{\sum_{b,t} m_{b,t} \sum_c w_c^{\text{pseudo}} + \epsilon} \tag{8}$$

3. **Date Features Integration**: The enhanced version incorporates Year and Day-of-Year. Year is mapped to a learnable Embedding; for Day-of-Year, sine and cosine components are calculated and fused:

$$h = h + \text{meta\_proj}([\text{Emb}_{\text{site}}, \text{Emb}_{\text{hour}}, \text{Emb}_{\text{year}}, \text{Proj}_{\text{doy}}]) \tag{9}$$

4. **Taxonomic Auxiliary & Perch Family Head**: Introduces a taxonomy-level auxiliary loss $\mathcal{L}_{\text{fam}}$ and a family teacher distillation head. The total task loss becomes:

$$\mathcal{L}_{\text{total\_enhanced}} = \mathcal{L}_{\text{BCE\_pseudo}} + \lambda_{\text{dist}}\mathcal{L}_{\text{dist\_masked}} + \lambda_{\text{fam}}\mathcal{L}_{\text{fam}} + \lambda_{\text{te\_fam}}\mathcal{L}_{\text{te\_fam}} \tag{10}$$

5. **StratifiedGroupKFold & Mixup**: The cross-validation split is upgraded to StratifiedGroupKFold (SGKF) based on species richness buckets. A Soft-label Mixup strategy is introduced during training (configuration parameter `mixup_alpha=2.0` represents the α parameter of the Beta distribution), performing consistent temporal interpolation of features and labels within batches.
6. **Unannotated Window Label Correction**: For 6 soundscapes audio files with incomplete annotations, we selected some windows with high confidence and applied minor corrections, modifying some unannotated windows to `nocall`.

3.4. ResidualSSM

As a lightweight temporal refinement network for the second pass, ResidualSSM aims to perform residual correction in the logit space based on the logits output by the first-pass ProtoSSM. The core architectural differences are as follows:

1. **Concat-Input Representation**: ProtoSSM only accepts the raw frame-level embedding vectors (dimension d_{input}). ResidualSSM employs concatenated dual-channel features as input, concatenating the raw embedding vectors with the logits generated in the first pass along the channel dimension (dimension $d_{\text{input}} + n_{\text{classes}}$).
2. **Lightweight Sequential Backbone**: ResidualSSM adopts an extremely lightweight single-layer Bidirectional Selective State Space structure ($d_{\text{model}} = 32$, $d_{\text{state}} = 4$), whereas ProtoSSM typically uses multiple layers and wider state space channels.
3. **Zero-Initialized Linear Head**: ResidualSSM directly employs a standard linear layer to output additive correction terms. The weights and biases of this linear head are initialized to zero:

$$\text{weight} \leftarrow 0, \quad \text{bias} \leftarrow 0 \tag{11}$$

4. **Logit-Space Regression Target**: The training target is the residual r between the first-pass predictions in the logit space and the ground truth labels:

$$r = \text{logit}(y_{\text{clipped}}) - \text{clip}(w_{\text{blend}} \cdot z_{\text{first\_pass}}, -50, 50) \tag{12}$$

   The loss function employs a regression loss combined with a temporal mask m:

$$\mathcal{L}_{\text{residual}} = \text{RegLoss}(\text{pred}, r) \cdot m \tag{13}$$

5. **Additive Logit Fusion**: During inference, the correction terms are incorporated additively:

$$\text{probs} = \sigma\left(z_{\text{first\_pass}} + \lambda_{\text{corr}} \cdot \text{corr}_{\text{ResidualSSM}}\right) \tag{14}$$

   where λ_{corr} is the correction step size coefficient, and σ is the Sigmoid activation function.

3.5. Postprocessing

Following the results of the residual correction model, the pipeline introduces a series of post-processing techniques in the probability space to further calibrate confidence:

1. **Per Taxonomy Temperature Scaling**: For the probabilities $p_{t,c}$ output after residual correction, they are first reverse-mapped back to the logit space, then class temperature parameters τ_c are applied for different taxonomic groups:

$$z_{t,c} = \log\left(\frac{p_{t,c}}{1-p_{t,c}}\right), \quad p'_{t,c} = \sigma\left(\frac{z_{t,c}}{\tau_c}\right) \tag{15}$$

   If class c belongs to a "texture" vocalization group (such as insects and amphibians), then $\tau_c = \tau_{\text{texture}} = 0.95$; otherwise, $\tau_c = \tau_{\text{non-texture}} = 1.10$.
2. **Taxonomy Smoothing**: To mitigate the model's confusion over closely related species, the predicted probabilities are smoothed toward the mean probabilities of the same Genus and Class using hyperparameters $\alpha_{\text{genus}} = 0.15$ and $\alpha_{\text{class}} = 0.05$:

$$p^{(1)}_{t,c} = (1-\alpha_{\text{genus}}) \cdot p'_{t,c} + \alpha_{\text{genus}} \cdot \frac{1}{|G_c|} \sum_{i \in G_c} p'_{t,i} \tag{16}$$

$$p^{(2)}_{t,c} = (1-\alpha_{\text{class}}) \cdot p^{(1)}_{t,c} + \alpha_{\text{class}} \cdot \frac{1}{|C_c|} \sum_{i \in C_c} p^{(1)}_{t,i} \tag{17}$$

3. **Temporal Smoothing**: For taxa with specific temporal vocalization patterns, smoothing across adjacent time windows ($t-1, t, t+1$) is applied. For linear taxa (e.g., continuous vocalizing insects), weighted average smoothing is adopted:

$$p^{(3)}_{t,c} = (1-\alpha_c) \cdot p^{(2)}_{t,c} + \alpha_c \cdot \frac{w_{t-1}p^{(2)}_{t-1,c} + w_{t+1}p^{(2)}_{t+1,c}}{w_{t-1} + w_{t+1}} \tag{18}$$

   For event taxa (e.g., short vocalizing birds), local maximum smoothing is adopted:

$$p^{(3)}_{t,c} = (1-\alpha_c) \cdot p^{(2)}_{t,c} + \alpha_c \cdot \max\left(p^{(2)}_{t-1,c}, p^{(2)}_{t,c}, p^{(2)}_{t+1,c}\right) \tag{19}$$

4. **Top-1 Power File-Level Scaling**: To amplify the confidence of highly certain species across the entire audio clip, we extract the mean of its Top-N (N=1) probabilities across all windows in the file, μ_c, and raise it to a power $\gamma = 0.75$ [4]:

$$\mu_c = \frac{1}{N}\sum_{k=1}^{N} p^{(3)}_{(k),c}, \quad S_c = (\mu_c)^{\gamma}, \quad p^{\text{final}}_{t,c} = \min\left(p^{(3)}_{t,c} \cdot S_c,\ 1.0\right) \tag{20}$$

4. Execution Process and Post-Competition Analysis

4.1. Overview of Execution During the Competition

Although the concept of class grouping only took shape mid-competition, for clarity of exposition, we will first define this concept and then review the training journey. Based on "whether 1-minute annotated training audio is available" and "whether Perch features exist," we divided all 234 classes into 4 groups:

Among them, Groups A and C refer to the unmapped classes lacking Perch features. Furthermore, during an in-depth analysis of Perch logits across the training soundscapes, we identified three amphibian classes in Group B (*23150*, *23154*, and *476521*) whose maximum and 99.9th percentile logits were constantly below -3.0 with extremely small variance (< 0.05). We defined these as "Poor Perch

Table 1
Class Grouping Strategy

Group	1-Min Audio	Perch Features	Classes
A	No	No	2
B	No	Yes	157
C	Yes	No	29
D	Yes	Yes	46

classes" and, due to their completely uninformative foundation features, applied the same suppression and fusion strategy to them as Group A.

We defined the pure version of ProtoSSM, without any post-processing or model fusion, as the baseline version. In terms of network structure, the core hyperparameters were set as follows: backbone dimension $d_{\text{model}} = 128$, SSM state dimension $d_{\text{state}} = 16$, stacked $n_{\text{ssm\_layers}} = 2$ layers, metadata fusion dimension meta_dim $= 16$, and dropout $= 0.15$. Model training was conducted based on 5-Fold Cross-Validation, with optimizer parameters: learning rate lr $=$ 8e-3, weight decay 2e-4, maximum training epochs $n_{\text{epochs}} = 150$, and early stopping patience $= 10$. Although we calculated CV Macro AUC on the 66 annotated 1-minute audio files, due to the scarcity of training samples and the fact that some long-tail classes only appeared in single audio files, CV Macro AUC fluctuated wildly and could only serve as an auxiliary reference; the primary evaluation metric still heavily relied on the Public LB.

In the early stages, feature extraction took too long because we used the native TensorFlow version of Perch V2 for CPU inference, resulting in multiple failed submissions due to exceeding the 90-minute limit. By introducing a forced timeout truncation mechanism, we barely obtained a Public LB of 0.901. Subsequently, we replaced the feature extractor with the ONNX version of Perch V2 [7]. Compared to the native TensorFlow SavedModel—where the default `serving_default` inference signature incurs severe graph execution and tensor conversion overheads—our pipeline encapsulates the ONNX model into a lightweight `infer_fn` wrapper[1]. This wrapper operates directly on native NumPy arrays via `onnxruntime`, entirely bypassing the TensorFlow bottlenecks. This architectural substitution achieved a roughly $4\times$ speedup for feature extraction on Kaggle's CPUs, compressing the entire notebook's execution time to under 25 minutes, and improving the baseline Public LB to 0.905.

Over the subsequent 40+ submission experiments, we enabled various enhancement strategies for ablation comparisons, during which the Public LB fluctuated between 0.895 and 0.910. Comparisons revealed that introducing Date features did not achieve the expected results, and early residual correction models did not yield ideal gains; only Mixup data augmentation brought stable score improvements. After increasing the `mixup_alpha` from 0.4 to 2.0 and stripping out date features, the Public LB touched 0.910, which we established as the New Baseline for ProtoSSM.

We then gradually introduced Temporal Smoothing and Top-N File-Level Scaling, pushing the Public LB to 0.913 and 0.919, respectively. Manual label correction on 5 environmental audio files with missing annotations caused the Public LB to surge to 0.923, though post-competition review indicated this mainly overfit the Public test set distribution (< 0.001 Private LB gain).

Facing the score bottleneck, we re-examined the ResidualSSM network. We aligned it to the same 5-fold cross-validation mode and switched its loss function to L1/Huber Loss. After increasing λ_{corr} from 0.1 to 1.0, the Public LB broke through to 0.925. We then addressed the 31 "Unmapped classes" by introducing a family-level proxy prediction head, pushing the Public LB to 0.929. Through further polishing, the single-model Public LB peaked at 0.930.

We then initiated ProtoSSM Stage 2 based on pseudo-labeling, pushing the Public LB to 0.934. However, as detailed in Section 4.2.1, this was a "beautiful mistake"—overfitting to pseudo-label noise caused the Private LB to drop to 0.929.

The Kaggle community then open-sourced an ONNX-based SED model (single-model Public LB 0.917) [13]. A simple fusion of SSM_Rank $\times\, 0.6 +$ SED_Rank $\times\, 0.4$ in the Rank space instantly soared

[1] https://github.com/DYF7812/BirdCLEF-2026/blob/main/scripts/bc2026_utils.py

the score from 0.934 to 0.945. We then deployed the refined late fusion scheme based on the A/B/C/D grouping strategy, stably lifting the Public LB to 0.947, and ultimately reaching 0.948 with some post-processings after fusion.

In the final sprint, open-source solutions emerged using taxonomy smoothing in the Rank space to break 0.950. However, we believe that executing taxonomic compensation in a Rank space that has lost probability density violates statistical common sense. We insisted on front-loading taxonomy smoothing into the Logit/probability space after ResidualSSM correction. Although slightly inferior on the Public LB, our final Private LB of 0.944 proved the correctness of this approach [14].

To provide clear evaluation provenance and address all tracked leaderboard metrics throughout this study, we consolidate the entire score progression into Table 2.

Table 2
Evaluation Provenance of Key Submissions. All scores use the Macro AUC metric. Official leaderboard entries were submitted under the team name 'DTeam'.

Submission Variant	Phase	Model Type	Public LB	Private LB	Rank/Notes
Native TF Perch (Timeout Truncated)	Official	Single	0.901	-	-
ONNX Perch Baseline	Official	Single	0.905	-	-
ProtoSSM New Baseline (Mixup 2.0)	Official	Single	0.910	-	-
+ Temporal Smoothing	Official	Single	0.913	-	-
+ Top-1 File-Level Scaling	Official	Single	0.919	-	-
+ Manual Label Correction	Official	Single	0.923	-	-
+ ResidualSSM ($\lambda_{\text{corr}} = 1.0$)	Official	Single	0.925	-	-
+ Unmapped Proxy Head	Official	Single	0.930	-	-
ProtoSSM Stage 2 (Pseudo-labels)	Official	Single	0.934	0.929	-
Community SED Model	Official	Single	0.917	-	-
Simple Fusion (SSM + SED)	Official	Ensemble	0.945	-	-
Grouped Late Fusion + Post-processing	Official	Ensemble	0.950	0.944	Silver (Rank 115)
ProtoSSM Stage 1 + SED (No Stage 2)	Post-comp	Ensemble	0.948	0.946	-

4.2. Post-Competition Ablation Study

4.2.1. The Pitfall of ProtoSSM Pseudo-labels

The Stage 2 model incorporating pseudo-labels actually performed worse on the Private LB than the pure version. In a post-competition ablation study, we completely stripped out the Stage 2 module, relying solely on ProtoSSM Stage 1 and SED for late fusion. This streamlined architecture reached 0.948 on the Public LB and secured 0.946 on the Private LB—a result that would have placed us even higher. This diagnostic exposed an important empirical lesson: while confidence-based pseudo-labeling is highly effective for CNNs (like SED), its naive application to our ProtoSSM pipeline failed in this specific extremely low-resource setting. We hypothesize that introducing noisy soft-distribution pseudo-labels without more sophisticated filtering caused the model to allocate its sequence-modeling capacity toward overfitting environmental background noise, rather than learning true temporal vocalization features. Therefore, we do not conclude that ProtoSSM is inherently incompatible with pseudo-labeling; rather, effectively harnessing unannotated data for such sequence architectures likely requires more rigorous experimental designs and advanced noise-resistant strategies.

4.2.2. Evaluation of ProtoSSM Enhancements and Post-processing

After the competition, we conducted a rigorous comparative evaluation of the core enhancement measures and post-processing modules. Due to strong non-linear synergies between components (for example, the true generalization benefits of Mixup only fully emerged after post-processing correctly calibrated the probability space), the impacts of single-model enhancements were primarily estimated via "leave-one-out" ablations from a fully optimized baseline, whereas the gains of fusion strategies

were derived by tracking sequential multi-model submissions. The table below details these synthesized ablation results (negative values indicate performance degradation):

Table 3
Ablation Results of Enhancement and Post-processing Strategies

Enhancement / Post-processing	Private LB Impact
Date Features Integration	-0.005
SGKF	< 0.001
Manual Label Correction	< 0.001
Mixup 2.0	$+0.004$
ResidualSSM	$+0.005$
Unmapped Taxonomic Distillation	< 0.001
Per Taxa Temperature Scaling	< 0.001
Taxonomy Smoothing	$+0.005$
Temporal Smoothing	$+0.004$
Top-1 Power File Level Scaling	$+0.011$
Ensemble of SED	$+0.010$
Heterogeneous Grouping Fusion	$+0.001$
Rank Space Post-processing	< 0.001

The data clearly reveals: **Date features** had significant negative impact (-0.005), confirming that hard-coding temporal information leads to overfitting. **SGKF** and manual **Label Correction** provided virtually no gain (< 0.001). The aggressive **Mixup 2.0** secured a notable improvement ($+0.004$), demonstrating robustness for long-tail classes. **ResidualSSM** contributed $+0.005$, confirming the validity of logit-space residual corrections. **Top-1 Power file-level scaling** generated the largest single-item gain ($+0.011$). The **SED Heterogeneous Ensemble** ($+0.010$) reiterates the dominant role of model diversity.

Given that feature-level Mixup exhibited astonishing regularization capabilities, we designed additional ablation experiments to explore its mathematical differences from traditional waveform-level Mixup:

Table 4
Feature-level vs. Waveform-level Mixup Comparison

λ	RMSE (Emb.)	RMSE (Logits)	Cosine (Emb.)	MAE (Prob.)
0.25	0.044	0.519	0.900	0.045
0.50	0.049	0.582	0.892	0.051
0.75	0.037	0.445	0.932	0.038

The data reveals that the non-linear deviation peaks at $\lambda \approx 0.5$ (Logits RMSE ≈ 0.58, MAE probability deviation ≈ 0.051). However, the Mean Cosine similarity on dense embedding vectors remains solidly above 0.89, definitively proving that executing low-cost linear Mixup in the pre-cached feature space closely approximates the physical effects of actual acoustic mixing.

4.2.3. ProtoSSM Model Diagnostics

Given that the ProtoSSM training set is extremely scarce and contains a large number of long-tail classes, the cross-validation Macro AUC often fluctuates wildly due to missing positive samples, making it difficult to serve as a reliable metric for evaluating the model's true generalization ability. Under a leaderboard lacking transparency, accurately determining ProtoSSM's true performance across various data groupings to allocate scientific weights for late fusion became a highly challenging technical blind spot. To address this, we used the opportunity for late submissions post-competition to design a black-box diagnostic system on the test set based on "forced probability truncation."

The core logic of our diagnosis utilizes a feature of the Macro AUC calculation mechanism where "all 0.5 predictions score 0.5." By enforcing a constant output of 0.5 on a specific grouping, we can back-calculate the true AUC of the remaining groupings. The specific experimental process is as follows: First, in a single-model submission, we hard-coded the predicted values of all 234 classes to 0.5, unsurprisingly obtaining a Private LB of 0.500, establishing a baseline. Subsequently, we completely stripped away all post-processing modules, Stage 2 pseudo-labels, and fusion with other heterogeneous models, retaining only the pure versions of ProtoSSM and ResidualSSM. Upon submission, we sequentially reset the prediction outputs of a specific test grouping (e.g., A/B/C/D) to 0.5 to observe the performance changes of the remaining groupings after that grouping was "masked." Furthermore, we submitted predictions from the pure Perch V2 foundation model on Groups B and D, aiming to use a control variable method to isolate and quantify the independent gains brought by "sequence modeling" in ProtoSSM compared to simple "feature distillation."

After obtaining the masked Private LB scores for each group, we attempted to back-calculate the true Macro AUC of that group based on the number of classes reset. However, when we naively assumed all 234 classes were "valid classes" (i.e., possessed at least one positive sample in the private test set) and substituted them into the back-calculation, a severe paradox emerged: under both the Perch baseline and ProtoSSM, the corrected AUC for Group D incredibly breached the theoretical ceiling of 1.0. This mathematical contradiction undeniably proved a crucial fact: in the final Private test set, there must exist a certain number of classes with absolutely no positive samples, which are thus automatically ignored during the backend Macro AUC calculation.

To correct this deviation, we needed to logically pinpoint the distribution of these "invalid classes." Since mask testing showed that when Group A (2 classes) was reset, the LB deviated from 0.5, it indicated Group A contained at least 1 valid positive-sample class. Considering that both Groups C and D had extremely abundant 1-minute long audio training data, we made an aggressive but reasonable prior assumption: **all test classes lacking positive samples are entirely distributed within Group B (total 157 classes), which is only supported by short audio**. To balance the equation, we assumed the true number of valid classes in the test set was 174 (meaning 60 classes in Group B did not participate in the AUC calculation due to having no positive samples).

Based on this assumption, we derived the following correction formulas (see Table 5 for computed results under each hypothesis):

- For Groups A/C/D: $\text{AUC}_{\text{corr}} = \frac{N_{\text{valid}} \cdot \text{LB} - (N_{\text{reset}} - N_{\text{B\_invalid}}) \cdot 0.5}{N_{\text{eval}}}$
- For Group B: $\text{AUC}_{\text{corr}} = \frac{N_{\text{valid}} \cdot \text{LB} - N_{\text{reset}} \cdot 0.5}{N_{\text{eval}} - N_{\text{B\_invalid}}}$

Table 5
ProtoSSM Black-Box Diagnostic Results

				AUC_{corr}		
Configuration	Private LB	N_{eval}	N_{reset}	N_{valid}=234	N_{valid}=174	N_{valid}=145
ProtoSSM for ALL	0.91498	234	0	0.915	0.915	0.915
All Reset	0.50000	234	0	0.500	0.500	0.500
Perch only for B	0.73542	157	77	0.851	0.922	$1.002^{\dagger}$
Perch only for D	0.60670	46	188	$1.043^{\dagger}$	0.904	0.836
ProtoSSM for A	0.50327	2	232	0.882	0.784	0.737
ProtoSSM for B	0.73543	157	77	0.851	0.922	$1.002^{\dagger}$
ProtoSSM for C	0.55795	29	205	0.967	0.848	0.789
ProtoSSM for D	0.61832	46	188	$1.102^{\dagger}$	0.948	0.873

† Exceeds theoretical ceiling of 1.0; indicates the assumption has collapsed for this group.

Under this extreme hypothesis framework of "all invalid classes belong to Group B," the more invalid classes assigned to Group B, the more the corrected AUCs for Groups A/C/D would be diluted downwards, while Group B's own corrected AUC would soar. Calculations showed that when the

assumed invalid classes reached 89 (meaning only 145 global valid classes remained), Group B's corrected AUC would approach an absurd 1.0 (as confirmed by the $N_{\text{valid}}=145$ column in Table 5). Therefore, it must be emphasized: setting 174 valid classes and attributing "all invalid classes to Group B" is merely a heuristic hypothesis we introduced to deconstruct the black box. Once Groups C or D also incorporate a significant number of classes without positive samples, the absolute numerical values of this back-calculation will shift significantly.

Although absolute numerical values harbor uncertainties, the relative performance differences still provide us with highly valuable physical insights:

- **Group B (Features present, no long audio)**: ProtoSSM (0.73543) astonishingly overlaps with the pure Perch baseline model's score (0.73542). This strongly suggests that in a training environment utterly devoid of 1-minute continuous context, ProtoSSM's sequence modeling capability is completely paralyzed, degrading straight into a shallow linear wrapper of the Perch feature extractor.
- **Group D (Features present, long audio present)**: ProtoSSM's performance significantly outperforms the pure Perch baseline model (corrected AUC advantage up to over 0.04). This proves that after obtaining abundant "temporal fuel," the Mamba architecture successfully couples local spatial physical features deeply with broad environmental temporal rhythms, unleashing powerful sequence inference capabilities.
- **Group C (No features, long audio present)**: ProtoSSM's performance is moderate. Under the assumption of all valid classes, its corrected AUC reaches 0.967; but even under the severe squeeze of a limit of 145 valid classes, it still holds a baseline of 0.789.
- **Group A (Double absence class)**: ProtoSSM performs irredeemably worst. This rigorous diagnostic result corroborates our strategic decision made during the competition's final sprint: during final fusion, ruthlessly compressing ProtoSSM's weight to 0.1 for Group A and the 3 other "Poor Perch" classes was absolutely wise.

5. Conclusions and Limitations

5.1. Conclusions

Through architectural evolution spanning the entire competition and meticulous post-competition diagnostics, we have distilled the following core conclusions:

- **Feature-level Mixup is Both Efficient and High-Fidelity**: Even when performing linear mixing in the pre-cached Perch Embeddings space, it maintains over 89% cosine similarity with true waveform-level mixing, substantially defusing the overfitting crisis for few-shot classes at extremely low computational overhead.
- **Spatial Features + Temporal Context**: Lightweight state space models (ProtoSSM and ResidualSSM) show immense potential for modeling long sequences in low-resource environmental audio. By introducing knowledge distillation, the network successfully fused Perch's single-frame spatial representations with Mamba's macroscopic temporal perception capabilities.
- **Adhering to the Post-Processing Philosophy of Probability Density**: Post-processing based on biological and acoustic physical priors must be executed upstream within Logit or probability spaces that possess probability density. Blindly performing smoothing in a Rank space that has lost information richness presents irreversible risks of distribution shift.
- **The Dominance of Heterogeneous Ensembles**: The complementary integration of heterogeneous models (e.g., CNN-SED) remains the most robust avenue to break single-model performance ceilings and resist distribution shifts in long-tail data.

In summary, our work demonstrates that standing on the prior shoulders of universal bioacoustic foundation models (Perch V2), requiring only extremely lightweight tailored sequence modeling (core

architecture training takes less than 2 minutes) coupled with intuitive Bayesian post-processing, one can build an environmental acoustic classification system with strong competitiveness at an astonishingly high compute-to-output ratio.

5.2. Limitations and Outlook

Objectively evaluating, this solution still possesses the following limitations needing refinement:

- **Fragile Local Evaluation System**: Constrained by an extremely scarce training set (only 66 long audio segments), the 5-fold CV Macro AUC oscillated violently. Our architectural decisions were forced to rely heavily on Public LB feedback, increasing the risk of overfitting to specific leaderboard distributions.
- **Unexhausted Architectural Potential**: We were unable to conduct sufficient ablation experiments on techniques like Stochastic Weight Averaging (SWA), Cross-modal Attention Fusion, and Test-Time Augmentation (TTA). The theoretical generalization ceiling of ProtoSSM remains unreached.

Acknowledgments

We sincerely thank the organizers of this BirdCLEF competition—the LifeCLEF organizing committee and the Kaggle team—for providing a preeminent competition platform and exceedingly precious bioacoustic datasets for global researchers. We extend our deepest gratitude to all members of the Kaggle community who selflessly open-sourced code and shared insights.

Declaration on Generative AI

During this competition and code development process, we utilized Cursor and Antigravity, two advanced AI IDE tools, and leveraged their onboard Large Language Models, including Gemini, Claude, and Composer, to assist with code generation, debugging, and data analysis tasks.

For highly complex architectural tasks, we adopted a "human-machine collaborative validation" mode—first having Cursor generate a preliminary implementation blueprint, then passing it to the Antigravity agent for cross-review and logical refinement. After manual finalization, it was handed back to the IDE for code execution. Primary code generation relied mostly on Cursor's Auto mode (Composer model at the core), while a few highly difficult logic refactorings were handled by Claude Sonnet 4.6. All final merged and submitted code passed strict manual testing and online evaluation, and errors in critical pathways were manually debugged and fixed.

The drafting of this paper also benefited from AI assistance. We relied on Gemini Pro for LaTeX formula formatting, grammatical polishing, and structuring of prose logic. All content in the final manuscript has been reviewed word-by-word by the authors to ensure all technical viewpoints and post-competition reflections transfer our original thoughts with full authenticity.

References

[1] S. Kahl, T. Denton, L. Sugai, L. Piatti, W. H. Arruda Moreno, M. M. Barbosa, M. Eibl, C. Z. Fieker, C. M. Garcia, D. L. Hokama Sousa, J. E. d. A. Júnior, R. C. Kridler, M. Lasseck, A. V. d. Melo, H. Müller, M. d. O. Neves, M. G. d. Reis, L. K. Sampaio Teles, K.-L. Schuchmann, J. L. M. M. Sugai, K. T. P. Azevedo, P. d. N. Lopes, M. I. Marques, H. Klinck, H. Glotin, H. Goëau, W.-P. Vellinga, R. Planqué, A. Joly, Overview of BirdCLEF+ 2026: Acoustic species identification in the Pantanal, South America, in: Working Notes of CLEF 2026 – Conference and Labs of the Evaluation Forum, 2026.

[2] L. Picek, L. Adam, S. Kahl, R. Bossy, L. Chrobak, H. Goëau, K. Papafitsoros, H. Klinck, W.-P. Vellinga, R. Planqué, T. Denton, K. Barnard, C. Nédellec, L. Deléger, M. Courtin, G. Martellucci, I. Moummad, F. Vinatier, P. Bonnet, A. Joly, Overview of LifeCLEF 2026: Ai challenges for biodiversity understanding and ecosystem management, in: International Conference of the Cross-Language Evaluation Forum for European Languages (CLEF), Springer, 2026.
[3] N. Babych, 1st place solution: Multi-iterative noisy student is all you need, 2025. URL: https://www.kaggle.com/competitions/birdclef-2025/writeups/nikita-babych-1st-place-solution-multi-iterative-n, kaggle.
[4] V. Sydorskyi, F. Gonçalves, Tackling domain shift in bird audio classification via transfer learning and semi-supervised distillation: A case study on BirdCLEF+ 2025, in: CLEF 2025 Working Notes, Madrid, Spain, 2025.
[5] N. Babych, 1st place solution: Noisy student meets distillation, 2026. URL: https://www.kaggle.com/competitions/birdclef-2026/writeups/1st-place-solution-noisy-student-meets-distillati, kaggle.
[6] kapenon, 3rd place solution, 2026. URL: https://www.kaggle.com/competitions/birdclef-2026/writeups/3rd-place-solution, kaggle.
[7] justinchuby, Perch-onnx, 2025. URL: https://huggingface.co/justinchuby/Perch-onnx.
[8] B. van Merriënboer, V. Dumoulin, J. Hamer, L. Harrell, A. Burns, T. Denton, Perch 2.0: The bittern lesson for bioacoustics, arXiv preprint arXiv:2508.04665 (2025).
[9] A. Gu, T. Dao, Mamba: Linear-time sequence modeling with selective state spaces, arXiv preprint arXiv:2312.00752 (2023).
[10] J. Liang, B. Meyer, I. N. Lee, T.-T. Do, Self-supervised learning for acoustic few-shot classification, arXiv preprint arXiv:2409.09647 (2024).
[11] C. Tang, S. Baskiyar, State space models for bioacoustics: A comparative evaluation with transformers, arXiv preprint arXiv:2512.03563 (2025).
[12] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, in: Advances in Neural Information Processing Systems, volume 30, 2017.
[13] T. Arrants, BC2026 distilled-SED, 2026. URL: https://www.kaggle.com/code/tuckerarrants/bc2026-distilled-sed.
[14] shane18, yf D., ProtoSSM + SED blend & why rank-space taxonomy smoothing is a trap, 2026. URL: https://www.kaggle.com/competitions/birdclef-2026/writeups/protossm-sed-blend-and-why-rank-space-taxonomy-smo, kaggle.
[15] P. Kamongi, pantanal-distill-birdclef2026, 2026. URL: https://www.kaggle.com/code/kamongi/pantanal-distill-birdclef2026.

A. System Configurations

To facilitate reproducibility and support the ablation studies discussed in Section 4.2.2, we detail the primary hyperparameters and configuration variables of our optimized ProtoSSM single model (Private LB ROC AUC 0.935) in Table 6.

The full default configuration is defined in `scripts/base_config.py`, while overriding experimental configurations are located in the `notebooks/bc2026-train-new.py` training script. Readers can fully reproduce our ablation experiments by modifying these listed parameters. Both scripts are available in our public GitHub repository.[2,3]

[2] https://github.com/DYF7812/BirdCLEF-2026/blob/main/scripts/base_config.py
[3] https://github.com/DYF7812/BirdCLEF-2026/blob/main/notebooks/bc2026-train-new.py

Table 6
Primary Configuration Hyperparameters for ProtoSSM

Configuration Item	Value	Description
General & Training Setup		
`CFG.use_date_features`	`False`	Disables date features (year, day of year).
`CFG.oof_splitter`	`"stratified_group"`	Uses Stratified Group K-Fold for CV.
`CFG.revised_slice_overrides`	`{...}`	Manual label corrections mapping specific audio slices to `"nocall"` or class IDs. Leave empty to disable.
Data Augmentation		
`ProtoCFG.mixup_enabled`	`True`	Enables Feature-level Mixup 2.0.
`ProtoCFG.mixup_alpha`	`2.0`	Alpha parameter for Beta distribution in Mixup.
Model Architecture		
`CFG.residual_ssm.enabled`	`True`	Enables the two-stage ResidualSSM network.
`CFG.residual_ssm.correction_weight`	`1.0`	Weight for ResidualSSM correction (λ_{corr}).
Unmapped Classes Distillation		
`ProtoCFG.use_perch_family_te_head`	`True`	Enables proxy prediction head for unmapped families.
`ProtoCFG.use_perch_family_te_distill`	`True`	Enables taxonomic distillation for unmapped classes.
`ProtoCFG.unmapped_force_proto_alpha`	`False`	If `True`, forces $\alpha = 1$ for unmapped classes during fusion.
Probability Space Post-Processing		
`CFG.use_temporal_prob_smooth`	`True`	Enables temporal window smoothing in probability space.
`CFG.smooth_linear_boundary_align_interior`	`True`	Dynamic alignment for boundary smoothing windows.
`CFG.ssm_tax_smoothing`	`True`	Enables taxonomy smoothing directly in SSM probability space.
`CFG.smooth_family_alpha`	`{...}`	Dictionary of taxonomy smoothing weights (e.g., Amphibia: 0.25, Aves: 0.15).
File-Level Scaling		
`CFG.post_process_mul`	`"topn"`	Strategy for file-level scaling.
`CFG.post_process_mul_topn`	`1`	Number of highest scoring segments considered per class.
`CFG.post_process_mul_power`	`0.75`	Exponential power applied to the Top-1 scaling factor.
Ensemble & Late Fusion		
`CFG.pseudo.enabled`	`False`	Disables Stage 2 (Pseudo-labeling).
`CFG.tax_smoothing_enable`	`False`	Disables taxonomy smoothing in Rank space after ensemble.
`CFG.cab.enabled`	`False`	Enable CATBoost fusion (supports dynamic weights).
`CFG.stack.sed_enable`	`False`	Enable SED ensemble (supports dynamic weights).