Partial-Label Evaluation of Prototype Networks with Temporal Aggregation in BirdCLEF+ 2026

Tsz Chai Alan Liu[1]

[1]*Sonoform Labs, Canada*

Abstract

BirdCLEF+ 2026 requires identifying 234 bird, amphibian, insect, mammal, and reptile species from Pantanal soundscape recordings, scored per five-second window by macro-averaged ROC-AUC. Our development label set covered only 71 of these species, so local validation could measure ranking on those 71 but was blind to the other 163. We ask how this partial coverage distorts model selection for prototype models, which match a Perch audio embedding to per-species learned prototypes. Holding a full BirdCLEF+ pipeline fixed and swapping only its prototype member, we compared the deployed temporal ProtoSSMv2 model, which aggregates across the full recording, against two single-window models that score each window independently. Local validation favored a single-window model, yet on the hidden 234-species test that model fell below even the Perch-only baseline while ProtoSSMv2 stayed best. Diagnostics attribute the single-window model's collapse to its degrading rankings on the 163 unlabelled species. A no-temporal-aggregation ablation lowers the deployed model from 0.94218 to 0.94089 private macro-AUC, showing that temporal aggregation is one source of the deployed model's advantage but not the only one, since the ablated model still beats the baselines. A matched-control study also confirms that prototype explanations can be class-pure and call-focused in a cleaner setting. Partial-label validation can therefore overrate window-level prototypes and hide the value of temporal aggregation in BirdCLEF-style soundscape systems. Our code is available at https://github.com/Sonoform-Labs/BirdClef.

Keywords

BirdCLEF+ 2026, Bioacoustics, Prototype Models, Temporal Aggregation, Partial-Label Evaluation, Soundscape Classification

1. Introduction

The BirdCLEF+ 2026 challenge [1], part of LifeCLEF 2026 [2], asks participants to identify 234 species from Brazilian Pantanal soundscapes, spanning birds, amphibians, insects, mammals, and reptiles. Systems output species-presence probabilities for each five-second window, and the official metric is macro-averaged ROC-AUC across target species. All inference must finish within a 90-minute CPU-only budget on a Kaggle notebook, which rules out large ensembles and favors reuse of pretrained representations.

This work concerns local validation, a practical obstacle in developing such systems. The hidden test spans a large multi-species label space, but the labelled soundscape data available during development covers only part of that label space. Our local set had positive labels for only 71 of the 234 species, and the remaining 163 could not contribute to local macro-AUC, so a model could raise its local score while degrading rankings on species that local validation cannot see.

We study this failure for prototype models. A prototype model scores a species by the similarity between an audio embedding and that species' learned prototypes, so its predictions trace back to learned acoustic exemplars rather than opaque logits. That interpretability makes prototype components attractive in bioacoustics, but it does not guarantee that a prototype model selected on local data transfers to the full task. Our starting point is a fixed BirdCLEF+ pipeline that fuses Perch scores, a prototype model, and an SED model through rank-space blending and post-processing, implemented from public Kaggle reference notebooks [3]. Its prototype member, ProtoSSMv2 [3], is temporal, aggregating information across the full one-minute recording. We compare it against simpler single-window prototype models that score each five-second window independently, holding every other pipeline

CLEF 2026 Working Notes, 21–24 September 2026, Jena, Germany

tcaliu@sonoform.org (T. C. A. Liu)

stage fixed.

Our hypothesis was that the head with the best local AUC would carry over to deployment. On the 71-species local set, the single-window prototype–Perch model achieved the best AUC, above both the Perch-only baseline and the temporal ProtoSSMv2 model, so under local model selection it was the natural choice. It did not carry over. On the hidden 234-species test the ordering inverted, the same model scored below the Perch-only baseline, and ProtoSSMv2 remained best. Diagnostics trace this reversal to coverage, since the single-window model degraded rankings most on the species with no local positives. A no-temporal-aggregation ablation of ProtoSSMv2 also lowered its hidden score, indicating that temporal aggregation contributes to ProtoSSMv2's advantage, though it is not the only difference between the models.

This working note makes four contributions. First, a controlled replacement protocol that swaps only the prototype member of a fixed Perch+prototype+SED pipeline, so hidden-test changes are attributable to the prototype model. Second, an offline interpretability-accuracy frontier for author-built prototype heads on frozen Perch embeddings. Third, hidden-test evidence that local validation with partial species coverage selects the prototype model that transfers worst. Fourth, a mechanism decomposition, including a temporal-module ablation and coverage-gating controls (using the prototype head only for species with local labels), showing that the failure concentrates on species absent from the local bank while the temporal model preserves signal there.

2. Related Work

Prototype networks make classification decisions interpretable by matching inputs to learned reference patterns. Where a standard classifier emits only class scores, a prototype model compares the input representation to learned prototype vectors and predicts from those similarities, yielding a "this looks like that" explanation grounded in representative learned examples [4]. In bioacoustics this is appealing because a species prediction can point to concrete acoustic evidence rather than an opaque model output.

Interpretability is not automatic, however. A learned prototype can capture background noise or compressed latent features instead of meaningful acoustic structure, so prototypes should be checked quantitatively, not trusted on sight [5]. We report prototype purity on the local set and, in a cleaner matched-control setting, a shortcut-rate check (how often a prototype's nearest patch lands on background rather than the target call), but our main question is whether a prototype model chosen by local validation stays useful inside the full pipeline.

Pretrained audio models underpin most recent bioacoustic systems. BirdNET [6], PANNs [7], BirdSet backbones [8], and Perch [9] supply strong representations for downstream classification. AudioProtoPNet [10] adapts the prototype paradigm to multi-label bird sound classification on a BirdSet-pretrained ConvNeXt backbone, showing that prototype explanations can be useful for bird-only classification. We keep Perch frozen, training the single-window prototype models on its embeddings, and one of them also blends prototype similarity with Perch's own species scores. This isolates the prototype model without retraining the backbone.

BirdCLEF systems typically combine several models with rank blending, sound-event detection, ecological priors, and temporal smoothing rather than relying on one classifier [11, 12]. Recent working notes emphasize transfer learning and semi-supervised distillation under domain shift and strict inference budgets [13, 14], but they report system-level performance without asking whether an offline ranking of an interpretable component survives insertion into the full system. We adopt that system-level setting deliberately, inserting each prototype model into the same pipeline and reading off its hidden-test score rather than judging it in isolation.

Table 1
Species distribution, approximate focal recording counts, and direct Perch logit coverage by taxon.

Taxon	Species	Focal rec.	Direct Perch
Aves	162	∼24,000	159
Amphibia	35	∼1,800	22
Insecta	28	∼900	13
Mammalia	8	∼300	6
Reptilia	1	∼20	0
Total	**234**	**∼27,000**	**203**

3. Task and Data

BirdCLEF+ 2026 targets acoustic species identification in Brazilian Pantanal soundscapes, extending previous bird-only editions [11, 12] to a broader multi-taxon setting. Each one-minute recording is scored in twelve five-second windows, and for each window a system outputs probabilities over 234 species. The official metric is macro-averaged ROC-AUC over species with positive examples in the hidden test.

Two difficulties shape the task. The first is domain shift. Focal training recordings are clean, single-species clips captured at close range, whereas the soundscape test windows come from passive monitoring devices and carry overlapping species (about 4.2 per positive window), rain and wind noise, and variable recording conditions. The second is class imbalance, shown in Table 1, where the most common birds have hundreds of focal recordings while several mammal and insect species have fewer than ten.

A separate coverage issue comes from the Perch class mapping. Of the 234 target species, 203 have direct Perch logit mappings, 3 are handled through taxonomic proxy mappings (borrowing the Perch logit of a taxonomically close species), and 28 produce no Perch logit signal, concentrated among insects and amphibians. This Perch gap is distinct from the offline evaluation gap that drives our main finding. The Perch gap concerns which species receive foundation-model support, whereas the evaluation gap concerns which species have positive labels in our local soundscape bank.

For local prototype development and offline evaluation, we use a labelled soundscape bank of 708 five-second windows from 59 one-minute source files. All folds are constructed at the source-file level using five-fold GroupKFold, so that all windows from a given recording fall entirely within one fold and no temporally adjacent windows leak across the split. Species enter offline macro-AUC only if they have at least one positive instance in the bank, yielding 71 evaluable species. The other 163 target species have no positive examples, so the offline endpoint can evaluate covered-species behavior but cannot evaluate whether a head damages ranking for absent species.

We report two endpoints. The offline endpoint is out-of-fold macro-AUC over the 71 covered species, used to compare prototype-head designs and construct the interpretability-accuracy frontier. The deployment endpoint is private hidden-test macro-AUC after replacing the prototype member inside the fixed core. Because Kaggle returns only aggregate public and private scores, with no per-example or per-species labels, we cannot compute paired tests, per-species confidence intervals, or p-values for leaderboard differences. Leaderboard results are therefore descriptive system-level outcomes, and the local bank is small enough that offline AUC should be read descriptively rather than as a locked benchmark.

4. Deployment Testbed

All replacement experiments run inside a fixed ProtoSSMv2+SED rank-fusion core adopted from public Kaggle notebooks [3]. The core has two active members, the deployed ProtoSSMv2 prototype member and a Perch-distilled SED member. Auxiliary EfficientNet members from the original pipeline are

Table 2
What is public versus this work. The deployed pipeline is a fixed, public testbed, and the contribution is the controlled replacement of its prototype member and the surrounding analysis.

Component	Source	Role in this study
Perch v2 embeddings / logits	public model [9]	fixed testbed
ProtoSSMv2 prototype member	public notebook [3]	fixed testbed (replaced)
Perch-distilled SED member	public notebook [3]	fixed testbed
Rank fusion + eco. priors	public notebook [3]	fixed testbed
Single-window prototype heads	**this work**	**intervention**
Replacement protocol + analysis	**this work**	**contribution**
Matched-control study (Sec. 8)	**this work**	**contribution**

disabled uniformly, so only the prototype member differs between submitted systems. Each member's scores are converted to per-species percentile ranks and weighted-summed. For member m, species c, raw score $s_c^{(m)}$, and fusion weight w_m,

$$r_c^{(m)} = \mathrm{rankpercentile}\big(s_c^{(m)}\big), \qquad S_c = \sum_m w_m \, r_c^{(m)}.$$

Rank-space blending is natural for macro-AUC, which depends only on within-species ordering. The deployed core scores 0.942 private macro-AUC and serves as the reference for all replacement comparisons.

The prototype member, ProtoSSMv2 [3], uses one class-specific prototype per species projected to 320 dimensions, computes cosine similarity to Perch-derived features, applies positive-temperature scaling (a strictly positive learned scale on the similarity), and blends prototype scores with Perch logits through a learned per-class gate $\gamma_c = \sigma(\alpha_c)$. A bidirectional residual selective state-space model [15, 16], a Mamba-style sequence model with input-dependent state updates, aggregates predictions across the twelve windows of each one-minute file, which is the temporal aggregation our single-window replacements lack. The SED member uses an EfficientNet [17] backbone distilled from Perch embeddings, and post-processing applies empirical-Bayes ecological priors and temporal smoothing, with the calibration-only steps verified to leave within-species rankings, and therefore macro-AUC, unchanged.

We claim no novelty for the base pipeline, which is a fixed deployment testbed. The contribution is the controlled replacement of its prototype member and the analysis around it, as Table 2 makes explicit.

5. Prototype Heads and Offline Evaluation

5.1. Replacement-head design

We train replacement prototype heads on frozen Perch v2 embeddings (1,536-d), and Perch itself is never retrained. Each head holds one or more learned prototypes $p_{c,k}$ per species c. With normalized embedding $z(x)$ and prototypes, the prototype activation and logit are

$$a_c(x) = \max_k \left\langle \tfrac{z(x)}{\|z(x)\|}, \tfrac{p_{c,k}}{\|p_{c,k}\|} \right\rangle, \qquad h_c(x) = \mathrm{softplus}(\eta_c)\, a_c(x) + b_c,$$

with learnable temperature η_c and bias b_c. A pure prototype head uses $h_c(x)$ directly. A fusion head instead combines the prototype logit with the Perch logit through a convex gate,

$$q_c(x) = \gamma_c(x)\, h_c(x) + \big(1 - \gamma_c(x)\big)\, \ell_c^{\mathrm{Perch}}(x),$$

where the single-window prototype model is a pure head and the single-window prototype–Perch model uses an input-dependent gate $\gamma_c(x)$. A static gate applies a per-class constant γ_c, whereas an input-dependent gate $\gamma_c(x)$ lets the prototype-versus-Perch mix vary with the input embedding. Training

Table 3
Stepwise build-up on frozen Perch embeddings. Δ values are relative to the previous row. The last two rows are the single-window prototype–Perch head (input gate) and the single-window prototype head (pure, high purity), and the deployed ProtoSSMv2 is shown for reference.

#	Config	AUC	Δ	Pur.	Δ
–	Perch alone	.739	–	–	–
1	Linear probe	.766	+.027	–	–
2	1-proto, no loss	.762	−.004	.69	–
3	+ cluster loss	.604	−.158	.944	+.254
4	+ separation loss	.605	+.001	.958	+.014
5	+ static gate	.742	+.137	.958	.000
6	+ input gate (proto–Perch)	**.830**	+.088	.437	−.521
7	pure prototype	.606	–	**.958**	–
–	ProtoSSMv2 (deployed)	.803	–	.790	–

uses FocalBCE loss [18], AdamW [19], and cosine annealing over 60 epochs. Interpretability variants add a cluster loss that pulls each prototype toward windows containing its class and a separation loss that pushes it from confirmed-absent windows. Prototype presence-purity is the fraction of prototypes whose nearest bank window (by cosine similarity) contains the prototype's assigned class.

5.2. The interpretability-accuracy frontier

Table 3 builds a head one component at a time on frozen Perch embeddings and reports covered-species out-of-fold AUC and purity.

Figure 1 plots the frontier, and three patterns define it. First, cluster loss is the interpretability lever, and adding it (rows 2 to 3) buys $+0.254$ purity at -0.158 AUC, the largest single effect in the table, while separation loss adds only $+0.014$ marginal purity. Second, the input-dependent gate trades purity for accuracy, and replacing the static gate (rows 5 to 6) gains $+0.088$ AUC at -0.521 purity by leaning on Perch logits and demoting the prototype's discriminative role. A "full-spec" head with both interpretability losses and the input gate collapses to 0.691 AUC, so no gated configuration in this sweep is both accurate and pure. Third, a static-gate fusion with both losses (row 5) is the only near-free-interpretability point, recovering Perch-level AUC (0.742) at 0.958 purity, though its absolute accuracy is far below the gated head. Increasing prototype count from one to three (row 7) does not help, so capacity is not the bottleneck, and the gated head peaks at 25 epochs before overfitting the 708-window bank.

The gated head's offline advantage over a linear probe is largest where Perch is weakest, at $+0.092$ macro-AUC on Aves and $+0.085$ on Amphibia, and near zero on Insecta ($+0.007$), where Perch embeddings are already discriminative. The pure prototype head's accuracy collapse is taxon-general.

5.3. Deployed-head audit

We audited the deployed ProtoSSMv2 to set the reference we aim to match. Its prototype purity is 0.789 (95% CI 0.69–0.87), far above a permutation null of 0.024 ± 0.018 (the purity expected when prototype-to-class assignments are shuffled), so its prototypes are valid by the presence-purity measure. Its per-class fusion gate is nearly uniform ($\gamma \approx 0.50$, standard deviation 0.003 across all 234 species), so the learned gate did not differentiate prototype versus Perch reliance, a negative result relative to its design motivation. Its per-taxon purity gradient (Insecta 0.96 $>$ Amphibia 0.88 $>$ Aves 0.64) runs opposite to AudioProtoPNet's bird-clean gradient. This contrast is consistent with prototype validity tracking acoustic distinctiveness, but backbone, training data, and prototype granularity all differ between the two settings, so we cannot attribute it to distinctiveness alone.

Under local validation the two single-window heads sit at opposite ends of a purity-accuracy trade-off. The pure prototype head reaches the highest purity (0.958) but the weakest AUC (0.606), while the

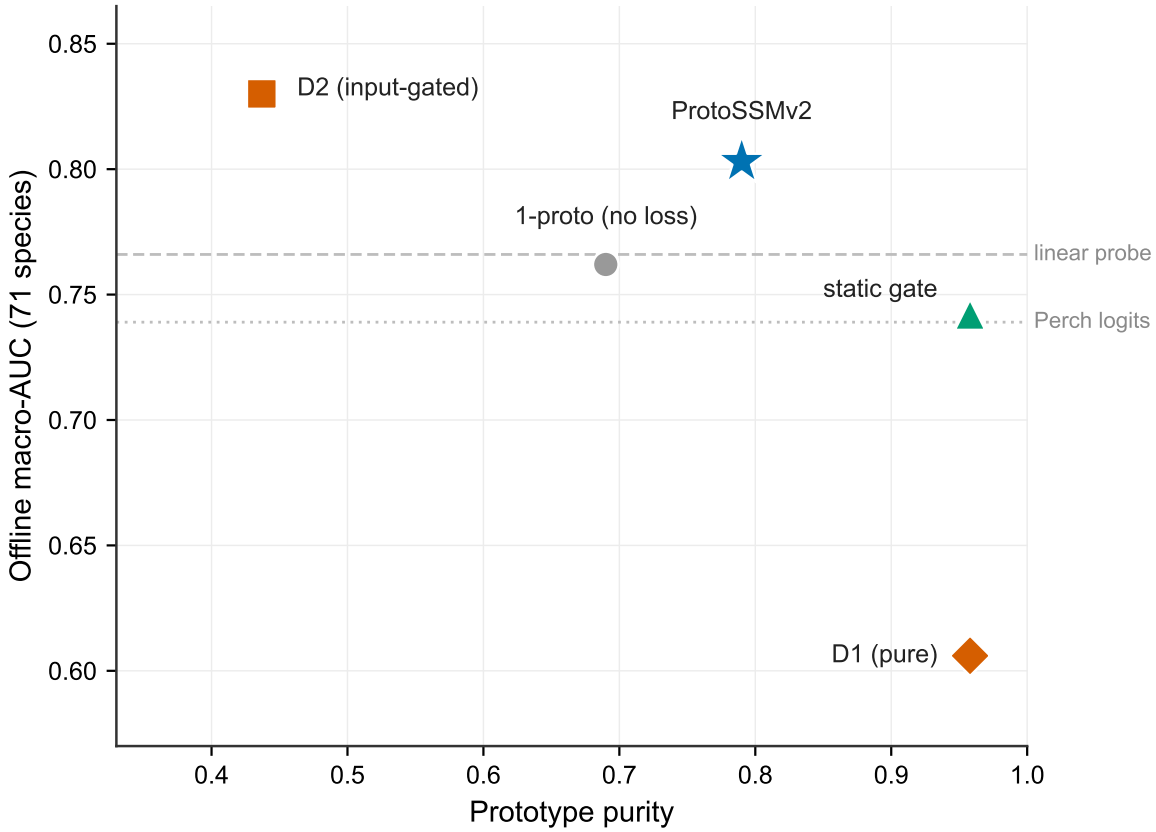

Figure 1: The interpretability-accuracy frontier. Each point is a head configuration from Table 3. Cluster loss raises purity at steep accuracy cost (rows 2 to 3), while the input-dependent gate recovers accuracy at the expense of purity (rows 5 to 6). The static-gate-plus-losses point (row 5) is the only near-free-interpretability design in the sweep, and the deployed ProtoSSMv2 sits in the interior of the frontier.

prototype–Perch head inverts this (0.437 purity, best AUC 0.830) by leaning on Perch. The deployed ProtoSSMv2 lands between them on AUC (0.803) and is not the local winner. Under local model selection, then, the prototype–Perch head would be chosen. The hidden test chooses differently.

6. The Offline Ranking Reverses

On the hidden leaderboard the local ranking inverts. Each prototype setting was swapped into the same fixed core and scored on the hidden test, and Table 4 lists every submitted variant so each score traces to one configuration, with per-submission metadata recorded in the released repository. Hidden labels are unavailable, so the rows are descriptive system-level outcomes.

The primary comparison is between the deployed ProtoSSMv2, the two single-window replacements, and the Perch-only baseline. ProtoSSMv2 has the highest private score (0.942), dropping the prototype member for raw Perch costs 0.003 (0.942 to 0.939), and both single-window replacements fall below that Perch-only floor of 0.939, the pure prototype at 0.938 and the prototype–Perch at 0.937. The offline ranking therefore does not transfer (Figure 2). The prototype–Perch head had the best offline AUC (0.830) but is the lowest primary replacement on the hidden test, while the pure prototype head had far lower offline AUC (0.606) yet a marginally higher private score. The gaps are small and rest on a single hidden test, but the direction holds across the public and private splits, so we read it as a system-level reversal rather than a single-row artifact.

The remaining rows are exploratory rescues and ablations. Removing the deployed head's temporal module lowers its score but leaves it above the Perch floor, coverage-gating the replacements recovers

Table 4
Hidden-test replacement results. Each row replaces or ablates the prototype member in the fixed core, and auxiliary CNN members are disabled uniformly. Off. is covered-species out-of-fold AUC, Pub. and Priv. are public and private hidden-test macro-AUC, and Δ is private minus the ProtoSSMv2 reference.

Prototype setting	Off.	Pub.	Priv.	Δ
Temporal ProtoSSMv2	.803	.949	**.942**	–
ProtoSSMv2, no temporal agg.	–	.948	.941	$-.001$
Prototype, coverage-gated	.606	.944	.940	$-.003$
ProtoSSMv2, coverage-gated	.803	.946	.940	$-.003$
Prototype–Perch, re-weighted	.830	.942	.939	$-.003$
Perch-only baseline	.739	.945	.939	$-.003$
Single-window prototype	.606	.938	.938	$-.004$
Prototype–Perch, coverage-gated	.830	.944	.938	$-.005$
Single-window prototype–Perch	**.830**	.941	.937	$-.005$
Prototype–Perch, broad focal	–	.939	.935	$-.007$

Table 5
Prototype–Perch gate and score diagnostics, split by species coverage. The head dampens Perch's discriminative signal even on the 163 species it never trained on. Variance ratio is $\mathrm{Var}(s^{\text{proto–Perch}})/\mathrm{Var}(s^{\text{Perch}})$ within each group.

Species group	Mean γ	ρ with Perch	Var. ratio
Covered by local labels (71)	0.779	0.319	0.485
Absent from local labels (163)	0.767	0.276	0.347

only to that floor, and broad focal training scores worst of all. The common thread is coverage, which the next section makes precise.

7. Mechanism Decomposition

The reversal follows from coverage. The pipeline scores all 234 species, but the prototype–Perch head was trained and selected on only 71. Table 5 splits the head's behavior by whether a species had local positives.

The input gate stays near $\gamma \approx 0.77$ even on the 163 absent species. For those species the prototypes received no positive training labels, so their scores carry little discriminative signal, and blending them in dilutes Perch's scores. Only about 35% of Perch's score variance survives for absent species, against 49% for covered species. Because macro-AUC weights species equally, this degradation across many absent classes becomes a hidden-test penalty that the covered-species offline endpoint cannot see. It also explains why the Perch-only baseline beat both single-window heads, since it left absent-species scores intact whereas the replacements altered them with no local signal to check the effect.

Coverage-gating. Using the prototype head only for covered species and raw Perch for the rest raised the pure prototype from 0.938 to 0.940 and the prototype–Perch from 0.937 to 0.938. Both moved toward the Perch floor ($\approx$0.939) but neither reached the deployed score, so absent-species damage explains part of the replacements' shortfall relative to the deployed score, but not all of it.

Temporal aggregation. Removing the deployed head's temporal SSM reduced private macro-AUC from 0.94218 to 0.94089, so temporal aggregation contributes measurably. But the ablated head still outranked the Perch floor (0.941 versus 0.939) and both replacements, so temporal modeling explains part of the deployed head's edge over the Perch floor, not all of it.

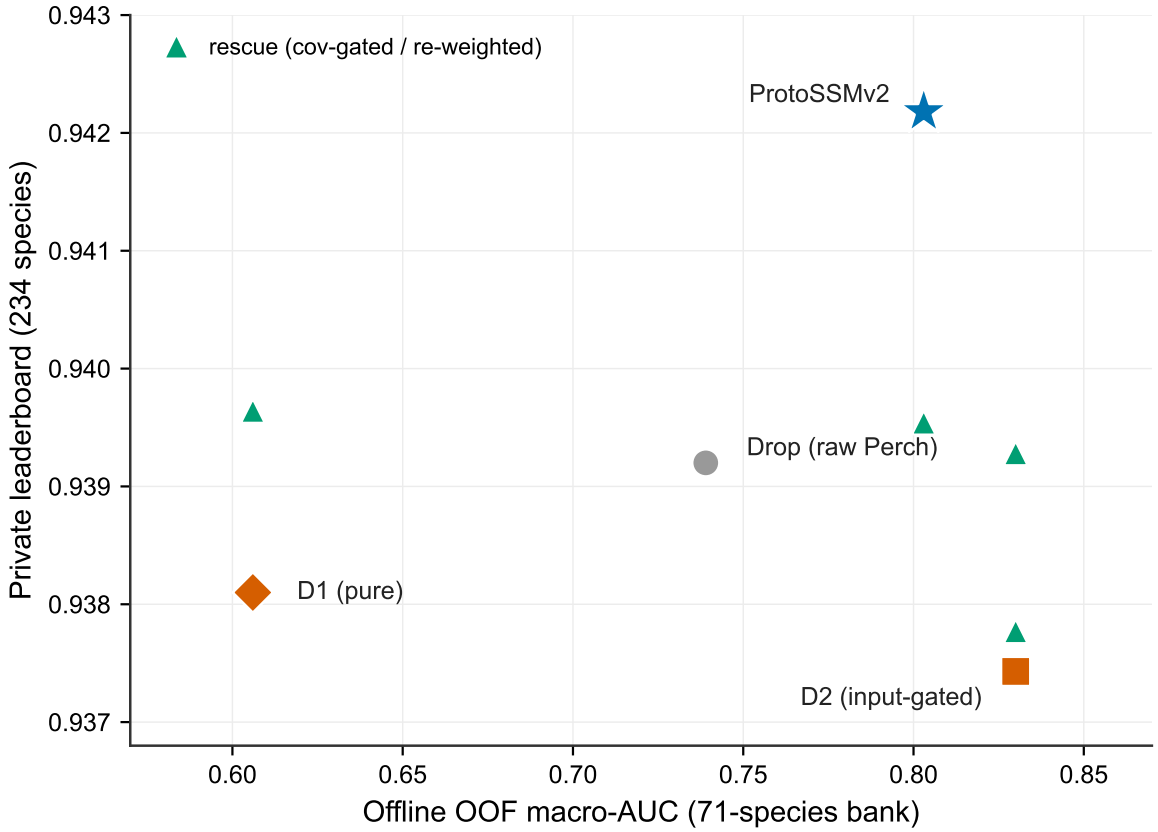

Figure 2: Offline out-of-fold AUC (x-axis) versus private leaderboard score (y-axis) for variants with an offline endpoint. Higher offline accuracy does not yield a higher deployment score. The single-window prototype–Perch model has the best offline AUC (0.830) but is among the lowest on the leaderboard, while the deployed ProtoSSMv2 sits at middling offline AUC yet ranks highest. The ranking reverses direction, and the absolute gaps are small (Section 9).

Signal preservation on absent species. Coverage-gating the deployed ProtoSSMv2 itself *hurt* it (0.942 to 0.940), the opposite of the replacements. Decomposing its $+0.003$ edge over raw Perch, only $+0.0003$ comes from covered species and $+0.0026$ from absent ones, so roughly 90% of its advantage is concentrated on the species it never saw training positives for. There the deployed head preserves more of Perch's discriminative signal, while the single-window heads dampen it. Its advantage over the Perch-only baseline therefore came mostly from the absent species rather than the 71 covered ones, plausibly because aggregating across the full recording lets evidence accumulate over time rather than being judged one window at a time.

Broad training backfires. We tested whether broader training coverage helps by training the prototype–Perch head on the 708 soundscape windows plus focal recordings spanning 206 species (5,090 windows), after verifying that our local Perch extraction reproduces the deployed one exactly (embedding correlation 1.000000). This variant scored worst of all (0.935), because the 5,090 focal windows outnumbered the 708 on-domain windows by about seven to one and dominated training, pulling the prototypes toward close-range, single-species focal acoustics, and whether balanced sampling or target-domain pseudo-labeling would avoid this is untested.

Together, these analyses point to a multi-causal gap. Coverage handling and temporal aggregation each contribute, and neither alone accounts for the deployed head's edge over the replacements.

These results frame a concrete trade-off between window-level and sequence-level prototype heads. Window-level heads are simpler and cheaper, they add no recurrent computation to the 90-minute

Table 6
Matched-control study. An AudioProtoPNet prototype head versus an identical sigmoid head on a shared ConvNeXt-Base backbone (BirdSet-XCL pretrained), five-fold, with only the classification head changed. Deltas are prototype minus sigmoid macro-AUC.

Quantity	Value
Backbone	ConvNeXt-Base, BirdSet-XCL pretrained
Cross-validation	5-fold
Training epochs	6 (original AudioProtoPNet uses 20)
Macro-AUC Δ (all taxa)	-0.007 ± 0.006 (95% CI spans 0)
Macro-AUC Δ (Aves)	-0.011 ± 0.001
Bird prototype purity	0.951 (permutation null 0.006)
Amphibian prototype purity	0.562
Insect purity coverage	3 of 28 species with enough clean clips

CPU budget, and on the 71 covered species the gated window-level head even led the offline AUC. Their limitation is structural, since scoring each five-second window in isolation prevents them from accumulating evidence across a recording, so a weak per-window signal, including from a species with no positive training labels, flows straight into the score without cross-window correction and degrades the absent species. The sequence-level ProtoSSMv2 pays for a bidirectional state-space pass over all twelve windows and is harder to read window by window, but that aggregation is what preserves Perch's signal on species the local bank never labelled. In this system the sequence-level cost buys robustness on the unmeasured majority, while the window-level simplicity comes at the price of that robustness.

8. Supporting Study: Prototype Interpretability in a Controlled Setting

The reversal concerns whether an interpretable component transfers to deployment, not whether prototypes can be valid at all. To check the latter in a clean setting, we compared a faithful AudioProtoPNet head [10] against an identical sigmoid baseline on a shared ConvNeXt-Base backbone pretrained on BirdSet-XCL [20, 8], using the same folds, inputs, optimizer, augmentations, and protocol so that only the classification head differs. Table 6 reports the setup and results.

The prototype head is accuracy-competitive, with a macro-AUC difference whose 95% confidence interval spans zero and only a small, consistent avian cost. Its prototypes are strongly class-pure for birds and call-focused rather than background-matching, with nearest patches that avoid the low-energy region at every tested cutoff, while amphibian prototypes are moderately pure and insect validity is coverage-limited. This validity result is narrow but important. Prototype explanations can be class-pure and call-focused in isolation, which is precisely why the deployment reversal is notable, since component validity did not predict deployment value.

9. Reproducibility and Limitations

Table 7 summarizes the setup. The deployment testbed is adopted unchanged from public Kaggle notebooks [3], and only the replacement heads and analyses are author-designed. The single-window heads are trained on frozen Perch v2 embeddings with FocalBCE, AdamW, and cosine annealing over 60 epochs (the gated head peaks near 25), under the same five-fold source-file GroupKFold as offline evaluation. Each leaderboard row is one submitted system generated by replacing the prototype artifact and leaving the rest of the core fixed. Code, evaluation notebooks, and per-row leaderboard metadata are released at https://github.com/Sonoform-Labs/BirdClef. Because hidden-test labels are not public, offline analyses are fully reproducible but leaderboard scores can be verified only through the original Kaggle evaluation environment.

Table 7
Main experimental setup.

Item	Setting
Local soundscape bank	708 labelled five-second windows
Source grouping	59 one-minute soundscape files
Validation split	5-fold GroupKFold by source file
Target species	234
Species with local positives	71 (163 absent)
Feature model	Frozen Perch v2 embeddings and logits
Compared prototype models	Temporal ProtoSSMv2, single-window prototype, single-window prototype–Perch, Perch-only baseline
Pipeline control	Perch, SED member, rank fusion, and post-processing held constant
Inference budget	90-minute CPU-only Kaggle notebook

Several limitations bound the results. The local bank is small and covers a minority of species, so offline AUC is a noisy, descriptive estimate rather than a secure ordering. Hidden scores are aggregates, so without per-species labels the mechanism analysis is diagnostic rather than a complete audit, and no paired tests are possible. The absolute leaderboard gaps are small, roughly 0.003 private AUC between ProtoSSMv2 and the Perch-only baseline and about 0.005 to the prototype–Perch head, and the temporal ablation accounts for only about 0.0013. The contribution is the direction of the offline-to-hidden reversal and its mechanism, not a general claim about prototype models. The deployed ProtoSSMv2 also differs from the single-window heads in architecture, capacity, and training as well as temporal context, so a fully matched comparison would need heads of matched capacity with and without temporal aggregation under the same training data, species coverage, and domain mix. Presence-purity is likewise a weak interpretability proxy, which is why the matched-control study adds a shortcut-rate check in a cleaner setting.

Read with those caveats, the result is practical for BirdCLEF-style development. When local labels cover only part of the target list, model selection can favor a component that does not transfer, and here that component was the single-window prototype–Perch head with the best local AUC. We recommend that component evaluations report label coverage, check behavior on absent classes with coverage-gated controls, and include at least one system-level replacement test rather than relying on a partial offline ranking alone.

10. Conclusion

We evaluated prototype models under partial-label validation in BirdCLEF+ 2026, holding a fixed Perch+prototype+SED pipeline constant and replacing only its prototype member. Local validation, covering 71 of 234 species, favored a single-window prototype–Perch model, but the hidden full-species test favored the temporal ProtoSSMv2 model, and the single-window model fell below even a no-prototype Perch baseline. A mechanism decomposition attributes the reversal to the replacement heads degrading rankings on the 163 species local labels could not measure, while the deployed head preserves foundation-model signal there. A no-temporal-aggregation ablation shows temporal aggregation contributes to but does not fully explain ProtoSSMv2's advantage, and a matched-control study shows prototype explanations can be valid in a cleaner setting. When a labelled soundscape set covers only part of the target species, local validation should be treated as a partial and possibly misleading guide to model selection, not a proxy for the full task.

Acknowledgments

We thank the BirdCLEF+ 2026 organizers and the LifeCLEF lab for hosting the competition and encouraging working-note submissions, and the Kaggle community, in particular the author of the public ProtoSSMv2 notebook [3] whose work forms the deployment testbed used here. We also thank the developers of the open-source tools used in this work, including PyTorch [21], Perch [9], BirdSet [8], and AudioProtoPNet [10].

Declaration on Generative AI

During the preparation of this work, the author used OpenAI's ChatGPT in order to draft the abstract, improve clarity within the writing, as well as LaTeX formatting. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

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] hideyukizushi, BirdCLEF+ 2026 — Perch v2 + ProtoSSM, Kaggle Notebook, 2026. URL: https://www.kaggle.com/code/hideyukizushi/birdclef-2026-perch-v2-protossm, accessed June 2026.

[4] C. Chen, O. Li, C. Tao, A. J. Barnett, J. K. Su, C. Rudin, This looks like that: Deep learning for interpretable image recognition, in: Advances in Neural Information Processing Systems 32 (NeurIPS), 2019, pp. 8928–8939.

[5] A. Hoffmann, C. Fanconi, R. Rade, J. Kohler, This looks like that… does it? shortcomings of latent space prototype interpretability in deep networks, in: ICML 2021 Workshop on Theoretic Foundation, Criticism, and Application Trend of Explainable AI, 2021. ArXiv:2105.02968.

[6] S. Kahl, C. M. Wood, M. Eibl, H. Klinck, BirdNET: A deep learning solution for avian diversity monitoring, Ecological Informatics 61 (2021) 101236.

[7] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, M. D. Plumbley, PANNs: Large-scale pretrained audio neural networks for audio pattern recognition, IEEE/ACM Trans. Audio, Speech, and Language Processing 28 (2020) 2880–2894.

[8] L. Rauch, R. Schwinger, M. Wirth, R. Heinrich, et al., BirdSet: A Large-Scale Dataset for Audio Classification in Avian Bioacoustics, 2024. arXiv:2403.10380.

[9] B. Ghani, T. Denton, S. Kahl, H. Klinck, Global birdsong embeddings enable superior transfer learning for bioacoustic classification, Scientific Reports 13 (2023) 22876.

[10] R. Heinrich, L. Rauch, B. Sick, C. Scholz, AudioProtoPNet: An interpretable deep learning model for bird sound classification, Ecological Informatics 87 (2025) 103081. doi:10.1016/j.ecoinf.2025.103081.

[11] S. Kahl, T. Denton, H. Klinck, et al., Overview of BirdCLEF 2024: Acoustic identification of under-studied bird species in the western ghats, in: Working Notes of CLEF 2024, CEUR Workshop Proceedings, 2024.

[12] J. S. Cañas, S. Kahl, T. Denton, et al., Overview of BirdCLEF+ 2025: Multi-taxonomic sound identification in the middle magdalena valley, colombia, in: Working Notes of CLEF 2025, CEUR Workshop Proceedings, 2025.
[13] A. Miyaguchi, M. Gustineli, A. Cheung, Distilling spectrograms into tokens: Fast and lightweight bioacoustic classification for BirdCLEF+ 2025, in: Working Notes of CLEF 2025, CEUR Workshop Proceedings, 2025.
[14] 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: Working Notes of CLEF 2025, CEUR Workshop Proceedings, 2025.
[15] A. Gu, T. Dao, Mamba: Linear-time sequence modeling with selective state spaces, in: Proc. ICML, 2024. ArXiv:2312.00752.
[16] A. Gu, K. Goel, C. Ré, Efficiently modeling long sequences with structured state spaces, in: Proc. ICLR, 2022.
[17] M. Tan, Q. Le, EfficientNet: Rethinking model scaling for convolutional neural networks, in: Proc. ICML, 2019, pp. 6105–6114.
[18] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proc. IEEE ICCV, 2017, pp. 2980–2988.
[19] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv preprint arXiv:1711.05101 (2019).
[20] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, A convnet for the 2020s, in: Proc. IEEE/CVF CVPR, 2022.
[21] A. Paszke, et al., PyTorch: An imperative style, high-performance deep learning library, Advances in Neural Information Processing Systems 32 (2019).