# Evaluating Impact of Distribution Shifts on Preprocessing and Modeling Choices for Speech Decoding from MEG Signals

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

We take machine learning approaches to the second phase of the LibriBrain Competition 2025, which requires phoneme classification from MEG recordings. We compare preprocessing strategies, phonological label representations, and model architectures. We highlight the impact of distribution shifts between validation and hidden evaluation data caused by inconsistent standardization, and find that simple residual convolutional networks outperform more complex architectures under this shift.

## 1 Introduction

Decoding speech units from non-invasive neuroimaging is a central goal for brain–computer interfaces [Lotte et al., 2018]. The LibriBrain 2025 phoneme classification track offers a standardized benchmark with aligned MEG segments and pre-defined data splits [Özdogan et al., 2025, Landau et al., 2025]. Our experiments revealed large gaps between validation and leaderboard metrics, highlighting unresolved distribution shifts.

Our contributions are threefold: (i) we quantify how group averaging and careful feature scaling stabilize training, (ii) we benchmark three neural backbones designed for short MEG windows, and (iii) we analyze failure modes caused by inconsistent standardization between validation and hidden evaluation data. These insights aim to guide future works on decoding of speech from MEG signals.

## 2 Related Works

### 2.1 Neural decoding from non-invasive brain signals

Non-invasive modalities like electroencephalography (EEG) and magnetoencephalography (MEG) are key to brain-computer interfaces (BCIs) and cognitive neuroscience, offering millisecond-level temporal precision despite noise and individual variability [Lotte et al., 2018, Schirrmeister et al., 2017].

Early research used system identification for stimulus reconstruction, e.g., predicting attended speakers from single-trial EEG [O'sullivan et al., 2015]. Recent advances combine self-supervised and contrastive learning to align neural signals with pretrained speech representations (e.g., wav2vec 2.0), achieving up to 41% accuracy across 1,000+ segments [Défossez et al., 2023]. MEG generally outperforms EEG, while functional near-infrared spectroscopy (fNIRS) and functional magnetic resonance imaging (fMRI) balance spatial resolution, latency, and portability [Cao et al., 2021].

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

### 2.2 Deep learning for time-series analysis

Deep networks extract nonlinear, hierarchical features from raw time series, replacing handcrafted engineering [Gamboa, 2017]. RNNs, LSTMs, and GRUs capture long-range dependencies for EEG classification and seizure prediction [Karim et al., 2017]. CNNs and temporal convolutional networks (TCNs) model local spatiotemporal patterns efficiently [Walther et al., 2023], while Transformers exploit self-attention for global dependencies. Hybrid CNN–LSTM and CNN–Transformer models combine both advantages, improving robustness [Sun et al., 2021]. These architectures have been adapted for multi-channel neural data with spatial filtering and frequency decomposition, supporting end-to-end decoding from sensors to cognition.

### 2.3 Speech and phoneme decoding from brain signals

EEG and MEG encode phonetic and speech representations. Early studies showed that MEG distinguishes phonetic contrasts and speech onset [Lukka et al., 2000], and later models decoded overt and imagined speech with over 90% accuracy under limited vocabularies [Dash et al., 2020, LaRocco et al., 2023]. Despite progress, most datasets remain small. New corpora such as *LibriBrain* and *SpeechImagery* enable cross-subject and cross-session generalization [Özdogan et al., 2025, Moreira et al., 2025]. Several works leverage phonological or articulatory features (e.g., place and manner of articulation) as intermediate targets, providing interpretable and biologically plausible representations Wang et al. [2012].

### 2.4 Domain adaptation and distribution shift

MEG decoding faces strong distribution shifts across sessions and subjects, degrading transfer performance. Solutions include (1) **knowledge distillation**, using teacher–student frameworks to align representations [Meng et al., 2019]; (2) **MMD alignment**, minimizing maximum mean discrepancy between source and target domains [Gretton et al., 2012]; (3) **data augmentation**, such as time–frequency perturbation and channel dropout; and (4) **adversarial or self-supervised learning**, enforcing domain-invariant priors [Lin and Zhang, 2023, Wang, 2025]. Together, these methods support robust and transferable neural decoding under realistic distribution shifts.

## 3 Methods

### 3.1 Dataset and preprocessing

Each example consists of a 500 ms window centered on a phoneme, recorded across 306 MEG channels in 250 Hz sample rate [Özdogan et al., 2025]. We adopt the competition's recommendation to average windows within each class. Unless noted, we form groups of size 100 and repeat the sampling procedure three times to balance denoising with data diversity.

**The train validation set is selected differently** from the PNPL library, as detailed in Appendix A. Both the train and validation sets are standardized with training statistics; the test set, containing the same examples as the validation set, is standardized with its own statistics to simulate the holdout set conditions, where input features are standardized per split. This mismatch creates a distribution shift analyzed in Section 5 and Appendix E.

### 3.2 Label representations

MEG labels are supplied as phoneme identifiers. We additionally map each phoneme to a 17-dimensional ternary vector of articulatory features derived from PanPhon [Mortensen et al., 2016]. From the original 21 features, we remove non-contrastive dimensions (e.g., "sg" for "spread glottis") to reduce redundancy. Optionally, we binarize the remaining values from $\{-1, 0, 1\}$ to $\{0, 1\}$ by treating $-1$ and 0 as absence of the feature. Table 1 illustrates the mapping for some phonemes. During training, the auxiliary regression target is optimized with mean-squared error; during inference, output is selected based on cosine similarity to the nearest phoneme.

Table 1: Example phonological vectors with their corresponding phonemes after removing non-contrastive features.

| Phoneme | syl | son | cons | cont | delrel | lat | nas | strid | voi | ant | cor | lab | hi | lo | back | round | tense |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| `/p/` | − | − | + | − | − | − | − | 0 | − | + | − | + | − | − | − | − | 0 |
| `/b/` | − | − | + | − | − | − | − | 0 | + | + | − | + | − | − | − | − | 0 |
| `/t/` | − | − | + | − | − | − | − | 0 | − | + | + | − | − | − | − | − | 0 |
| `/d/` | − | − | + | − | − | − | − | 0 | + | + | + | − | − | − | − | − | 0 |

### 3.3 Data augmentation

To mitigate class imbalance and sensor variability, we rely on lightweight augmentations applied randomly: additive Gaussian noise, temporal shifts, temporal masking, channel dropout, amplitude scaling, and frequency band perturbation. Detailed hyperparameters are provided in Appendix B.

### 3.4 Distribution alignment

We experiment with a small fully connected mapper trained with maximum mean discrepancy (MMD) to align standardized-then-averaged test features to the training distribution [Gretton et al., 2012]. The MMD is computed between source features $X$ and target features $Y$ as:

$$\mathrm{MMD}(X, Y) = \frac{1}{n^2} \sum_{i,j} k(x_i, x_j) + \frac{1}{m^2} \sum_{i,j} k(y_i, y_j) - \frac{2}{nm} \sum_{i,j} k(x_i, y_j),$$

The mapper $f_\theta$ is composed of three linear layers (input and output dimensions $306 \times 125$, hidden dimensions $512$) with ReLU activations. Training minimizes MMD with a Gaussian kernel of bandwidth 10 using Adam (learning rate $10^{-4}$, batch size 128, 50 epochs), where source features are drawn from the training set and target features from the test set after standardization and group averaging.

Appendix C visualizes a principal component projection of source, mapped, and target features. While the mapped features match the target mean, they collapse variance, explaining the reduced discrimination observed on the leaderboard.

## 4 Model architectures

**ResNet-style CNN.** Our primary model is based on the baseline model provided by Özdogan et al. [2025], which stacks temporal convolutions with residual connections and group normalization . A lightweight classifier head predicts phoneme logits and optional phonological features.

**STFT CNN.** We apply a short-time Fourier transform (STFT) to each channel (window size 25, hop 5) and share a 2D CNN across channels. The branch is trained jointly with the time-domain CNN but performs worse when evaluated on the hidden set.

**CNN-Transformer hybrid.** We append a 4-layer Transformer encoder with 8 attention heads to capture long-range dependencies across sensors and time. The module improves validation accuracy but amplifies overfitting when the evaluation statistics differ from training data.

Detailed model diagrams are provided in Appendix D.

## 5 Results

Table 2 summarizes representative submissions. All models are implemented in PyTorch, trained using a single RTX 5090 GPU by 10 epochs, and use group averaging with batches of 100 windows. Extended results with ablations are provided in Appendix E.

Some configurations are not submitted to the holdout set, denoted by dashes. The PanPhon feature experiments omit training F1-macro because optimization happens on continuous articulatory vectors with an MSE loss. Producing phoneme predictions requires a nearest-neighbor projection back into the discrete inventory, which we only execute on validation and leaderboard splits to avoid repeatedly decoding the entire training corpus each epoch.

Table 2: Representative F1-macro scores (%) for LibriBrain phoneme classification. Aug. denotes stochastic augmentations.

| Configuration | Train | Validation | Test | Holdout | Aug. | Notes |
|---|---|---|---|---|---|---|
| CNN + label-balancing | 90.81 | 71.95 | 47.47 | 35.40 | No | Repeated grouping $\times 10$ |
| CNN + LayerNorm | 96.62 | 44.49 | 43.17 | 24.40 | No | Layer normalization |
| CNN baseline | 34.55 | 45.08 | 39.53 | 13.20 | No | No group averaging |
| CNN + Augmentation | 48.55 | 49.31 | 34.03 | 18.80 | Yes | As in Section 3.3 |
| CNN-Transformer | 85.83 | 68.02 | 30.70 | 3.90 | No | |
| STFT CNN | 62.63 | 43.62 | 15.91 | – | No | STFT ($N_{\text{fft}} = 25, H = 5$) |
| CNN-Transformer | – | 51.68 | 3.33 | – | No | Ternary PanPhon |
| CNN baseline + Dist. Mapper | – | – | 0.06 | – | No | |

Simple CNNs with label balancing and group averaging achieve the best generalization; the performance gap widens when models depend on fine-grained normalization such as the CNN-Transformer hybrid. Adding phonological supervision yields minor validation gains but performs poorly on the test set, suggesting failure to generalize under distribution shift. Similarly, STFT preprocessing improves validation but degrades test performance. However, it remains unknown whether CNN variants with phonological targets would outperform time-domain models if evaluated on the holdout set.

Train-to-validation gaps highlight generalization ability. CNN with layer normalization achieves 96.62% train F1 but drops to 44.49% on validation, indicating overfitting. Label-balanced CNN maintains 71.95% validation F1 from 90.81% training, showing better robustness. Ill-formed normalization layers may overfit training statistics, exacerbating domain shifts. CNN-Transformer hybrid retains 68.02% validation F1 from 85.83% training, suggesting attention mechanisms capture more invariant features.

Validation-to-test comparisons reveal the impact of misaligned standardization. All models we've evaluated show significant drops from validation to test. In comparison, test-to-holdout gaps are fairly consistent (around 20%) across configurations, suggesting that the primary distribution shift arises from differing standardization statistics rather than other latent factors. Attempts to compensate with the distribution mapper (Section 3.4) improved qualitative alignment but reduced class separability, offering limited practical benefit.

# 6 Discussion and conclusion

**Contributions.** Our study highlights that how we aggregate MEG signals matters more than architectural novelty: label-balanced, group-averaged CNNs retain two-thirds of their training F1 on the leaderboard, while more expressive models collapse once the statistic mismatch is introduced. The most reliable recipe couples conservative preprocessing with restrained capacity, indicating that consistency across splits is a stronger signal of leaderboard success than raw training accuracy.

**Limitations.** Although it's tempting to attribute performance drops solely to distribution shifts, the reality is more complex. Factors such as model capacity, training dynamics, and the inherent variability of MEG signals all play a role.

Also, it remains unclear how well our findings generalize to other well-performed configurations and preprocessing pipelines from other teams. Future work should systematically evaluate a broader range of architectures and training strategies under controlled distribution shifts to validate the robustness of our conclusions.

**Future directions.** For future progress, combining distribution-aware normalization, adaptation methods that preserve intra-class spread, and supervision that retains the richness of phonological features while remaining compatible with the leaderboard metric may yield more robust speech decoding from MEG signals. Additionally, exploring self-supervised pretraining on large-scale unlabeled MEG data could help models learn invariant representations that generalize better across sessions and subjects.

# References


L. Cao, D. Huang, Y. Zhang, X. Jiang, and Y. Chen. Brain decoding using fnirs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12602–12611, 2021.

D. Dash, P. Ferrari, and J. Wang. Decoding imagined and spoken phrases from non-invasive neural (meg) signals. *Frontiers in neuroscience*, 14:290, 2020.

A. Défossez, C. Caucheteux, J. Rapin, O. Kabeli, and J.-R. King. Decoding speech perception from non-invasive brain recordings. *Nature Machine Intelligence*, 5(10):1097–1107, 2023.

J. C. B. Gamboa. Deep learning for time-series analysis. *arXiv preprint arXiv:1701.01887*, 2017.

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012.

F. Karim, S. Majumdar, H. Darabi, and S. Chen. Lstm fully convolutional networks for time series classification. *IEEE access*, 6:1662–1669, 2017.

G. Landau, M. Özdogan, G. Elvers, F. Mantegna, P. Somaiya, D. Jayalath, L. Kurth, T. Kwon, B. Shillingford, G. Farquhar, M. Jiang, K. Jerbi, H. Abdelhedi, Y. Mantilla Ramos, C. Gulcehre, M. Woolrich, N. Voets, and O. Parker Jones. The 2025 pnpl competition: Speech detection and phoneme classification in the libribrain dataset. *NeurIPS Competition Track*, 2025.

J. LaRocco, Q. Tahmina, S. Lecian, J. Moore, C. Helbig, and S. Gupta. Evaluation of an english language phoneme-based imagined speech brain computer interface with low-cost electroencephalography. *Frontiers in neuroinformatics*, 17:1306277, 2023.

G. Lin and J. Zhang. Multi-subdomain adversarial network for cross-subject eeg-based emotion recognition. *arXiv preprint arXiv:2308.14059*, 2023. URL `https://arxiv.org/abs/2308.14059`.

F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, and F. Yger. A review of classification algorithms for eeg-based brain–computer interfaces: a 10 year update. *Journal of neural engineering*, 15(3):031005, 2018.

T. Lukka, B. Schoner, and A. Marantz. Phoneme discrimination from meg data. *Neurocomputing*, 31(1): 153–165, 2000. ISSN 0925-2312. doi: https://doi.org/10.1016/S0925-2312(99)00178-2. URL `https://www.sciencedirect.com/science/article/pii/S0925231299001782`.

Z. Meng, J. Li, Y. Gaur, and Y. Gong. Domain adaptation via teacher-student learning for end-to-end speech recognition. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019. doi: 10.1109/ASRU46091.2019.9003811. URL `https://ieeexplore.ieee.org/document/9003811`.

J. P. C. Moreira, V. R. Carvalho, E. M. A. M. Mendes, A. Fallah, T. J. Sejnowski, C. Lainscsek, and L. Comstock. An open-access eeg dataset for speech decoding: Exploring the role of articulation and coarticulation. *Scientific Data*, 12(1):1017, 2025.

D. R. Mortensen, P. Littell, A. Bharadwaj, K. Goyal, C. Dyer, and L. S. Levin. Panphon: A resource for mapping IPA segments to articulatory feature vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484. ACL, 2016.

J. A. O'sullivan, A. J. Power, N. Mesgarani, S. Rajaram, J. J. Foxe, B. G. Shinn-Cunningham, M. Slaney, S. A. Shamma, and E. C. Lalor. Attentional selection in a cocktail party environment can be decoded from single-trial eeg. *Cerebral cortex*, 25(7):1697–1706, 2015.

R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball. Deep learning with convolutional neural networks for eeg decoding and visualization. *Human brain mapping*, 38(11):5391–5420, 2017.

J. Sun, J. Xie, and H. Zhou. Eeg classification with transformer-based models. In *2021 ieee 3rd global conference on life sciences and technologies (lifetech)*, pages 92–93. IEEE, 2021.

D. Walther, J. Viehweg, J. Haueisen, and P. Mäder. A systematic comparison of deep learning methods for eeg time series analysis. *Frontiers in Neuroinformatics*, 17:1067095, 2023.

H. Wang. Mmoc: Self-supervised framework with multi-model online collaboration for eeg emotion recognition. *arXiv preprint arXiv:2507.03977*, 2025. URL `https://arxiv.org/abs/2507.03977`.

R. Wang, M. Perreau-Guimaraes, C. Carvalhaes, and P. Suppes. Using phase to recognize english phonemes and their distinctive features in the brain. *Proceedings of the National Academy of Sciences*, 109(50):20685–20690, 2012.

M. Özdogan, G. Landau, G. Elvers, D. Jayalath, P. Somaiya, F. Mantegna, M. Woolrich, and O. Parker Jones. Libribrain: Over 50 hours of within-subject meg to improve speech decoding methods at scale. *arXiv preprint arXiv:2506.02098*, 2025.

# A Dataset details

Our dataset splits differ from those in the PNPL library. Table 3 summarizes the number of trials, phoneme counts, and standardization parameters for each split. The validation set for this competition phase is composed of 14 specific trials, listed in Table 4 [Landau et al., 2025]. The test set contains the same trials but is standardized with its own statistics to simulate the holdout set conditions.

Table 3: Summary of LibriBrain phoneme classification dataset splits.

| **Split** | **Trials** | **Phonemes** | **Standardization** |
|---|---|---|---|
| Train | 76 | 1,336,606 | Train statistics |
| Validation | 14 | 283,690 | Train statistics |
| Test | 14 | 283,690 | Test statistics |
| Holdout | 1 | 2,382 | Holdout statistics |
| **Total** | 91 | 1,622,678 | – |

Table 4: Validation trials for the dataset splits.

| **Subject** | **Session** | **Task** | **Trial** |
|---|---|---|---|
| 0 | 11 | Sherlock1 | 2 |
| 0 | 12 | Sherlock1 | 2 |
| 0 | 11 | Sherlock2 | 1 |
| 0 | 12 | Sherlock2 | 1 |
| 0 | 11 | Sherlock3 | 1 |
| 0 | 12 | Sherlock3 | 1 |
| 0 | 11 | Sherlock4 | 1 |
| 0 | 12 | Sherlock4 | 1 |
| 0 | 14 | Sherlock5 | 1 |
| 0 | 15 | Sherlock5 | 1 |
| 0 | 13 | Sherlock6 | 1 |
| 0 | 14 | Sherlock6 | 1 |
| 0 | 13 | Sherlock7 | 1 |
| 0 | 14 | Sherlock7 | 1 |

# B Augmentation recipes

For reproducibility we detail the augmentation hyperparameters:

- Gaussian noise with standard deviation 0.01 relative to the sample standard deviation.
- Temporal shift uniformly sampled in $\pm 40$ ms with circular padding.
- Temporal masking zeros out at most 80 ms of contiguous samples.
- Channel dropout randomly zeros 10% of sensors per step.
- Amplitude scaling multiplies the waveform by a factor drawn from $\mathcal{U}(0.9, 1.1)$.
- Frequency band perturbation performs a Fourier transform of the input signal in 100 Hz, randomly selects one or more frequency bands from the spectrum, and scales their amplitudes by a random 0.8 to 1.2 factor within a specified range to emulate varying spectral conditions. The modified spectrum is then transformed back into the time domain.
- Each augmentation is applied independently with probability 0.3.

# C Distribution alignment

The distribution shift between each trials is visualized as in Figure 1, where the pairwise channel mean cosine similarities are computed with raw features without standardization or group averaging.

While some pairs of trials show high similarity (e.g. Sherlock5, Session 11 to 13), most trials differ significantly with cosine similarities around 0.5. This variance likely contributes to the distribution shift observed between validation and hidden evaluation data.

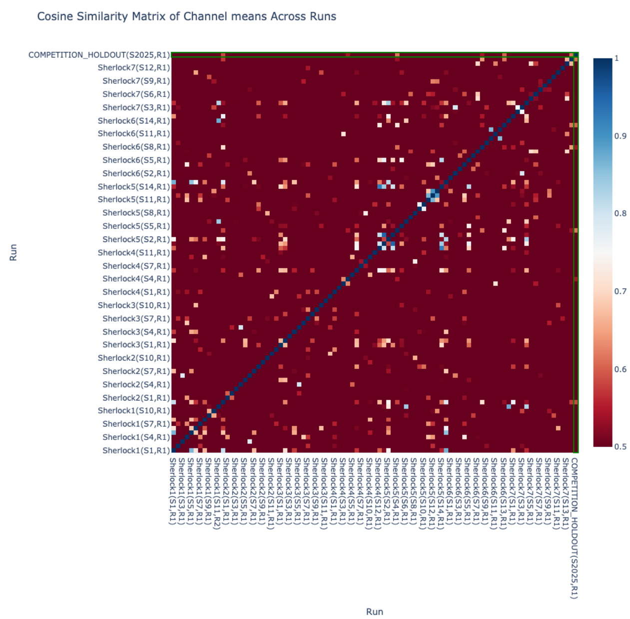


Figure 1: Pairwise cosine similarities between trials based on channel means. Trials used in holdout are highlighted with green boxes.

Figure 2 visualizes a PCA projection of source, mapped, and target features after training the MMD-based mapper. While the mapped features align with the target mean, they collapse variance, explaining the reduced discrimination observed on the leaderboard.

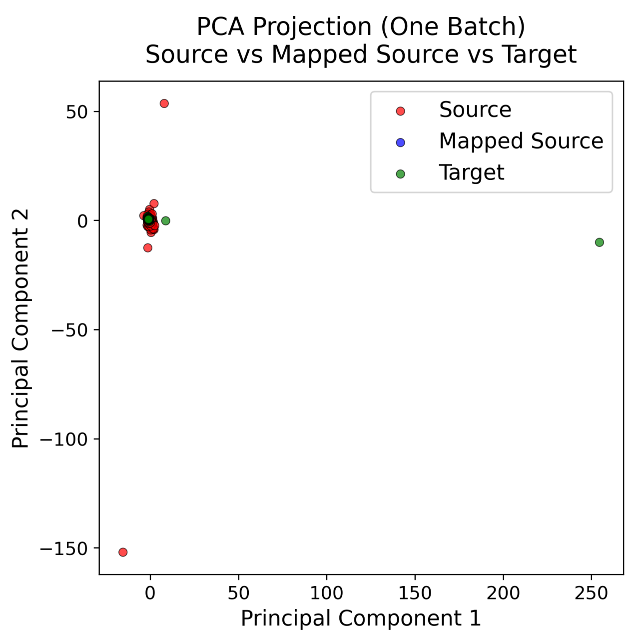


Figure 2: PCA projection of one batch where the mapper pulls source features (red) toward the target distribution (green) but compresses variance.

# D Detailed model configurations

Figure 3 illustrates the three model architectures evaluated. The CNN backbone (a) mainly consists of several 1D convolutional blocks with channel size $D = 256$ and 2 residual connections. An optional normalization layer (layer norm or batch norm) is added as shown in (a). The STFT-CNN model (b) applies STFT to each channel independently before feeding into a ResNet-based architecture with 2D convolutions. The CNN-Transformer hybrid (c) appends a 4-layer Transformer encoder after a CNN "filtering" stage, attempting to capture long-range dependencies across sensors and time.

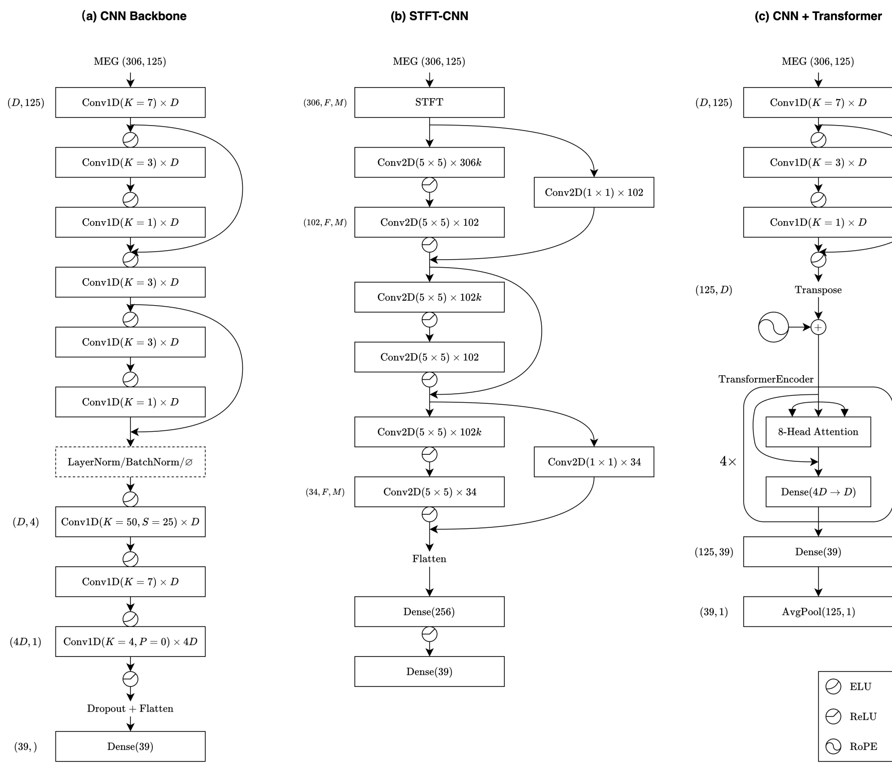


Figure 3: Overall model architecture. For the convolutional blocks, the stride $S$ is 1 and padding $P$ is set to maintain the same temporal dimension if not specified. (a) shows the basic CNN backbone, with optional layer norm or batch norm layer. (b) shows the STFT-CNN architecture, which applies STFT to each channel independently before feeding into the specialized ResNet-based model. (c) shows the CNN-Transformer architecture, which has a 4-layer Transformer encoder after several convolutional layers.

# E Extended results and visualizations

Table 5 reports the full leaderboard history. All models use group averaging with window size 100. "Repeat" specifies the number of iterations through the dataset that is then group-averaged. Additionally, the 2nd configuration in Table 2 is used to generate the confusion matrix in Figure **??**.

Figure 4 shows confusion matrices for the best-performing validation and leaderboard submissions. The same model sharply degrades on the leaderboard, confirming the distribution shift.

Table 5: Extended comparison of model and data configurations. Performance is reported in F1-macro (%). A dash denotes runs not submitted to the holdout set.

| Train | Validation | Test | Holdout | Backbone | Group Avg. | Label Bal. | Repeat | Note |
|---|---|---|---|---|---|---|---|---|
| 68.16 | 64.27 | 47.74 | 3.60 | CNN | Yes | Yes | 1 | +Augmentation |
| 90.81 | 71.95 | 47.47 | 35.40 | CNN | Yes | Yes | 10 | |
| 78.42 | 70.09 | 45.02 | – | CNN | Yes | Yes | 3 | |
| 80.06 | 71.77 | 44.28 | 21.60 | CNN | Yes | Yes | 1 | |
| 96.62 | 44.49 | 43.17 | 24.40 | CNN + LayerNorm | Yes | Yes | 1 | |
| 91.50 | 44.31 | 42.84 | – | CNN + BatchNorm | Yes | Yes | 1 | |
| 34.55 | 45.08 | 39.53 | 13.20 | CNN | No | No | 1 | |
| 48.55 | 49.31 | 34.03 | 18.80 | CNN | Yes | No | 1 | +Augmentation |
| 85.83 | 68.02 | 30.70 | 3.90 | CNN + Transformer | Yes | Yes | 1 | |
| 62.63 | 43.62 | 15.91 | – | STFT CNN | Yes | Yes | 5 | STFT, $N_{\text{fft}} = 25$ |
| 62.05 | 40.00 | 15.78 | – | STFT CNN | Yes | Yes | 5 | STFT, $N_{\text{fft}} = 50$ |
| 41.97 | 30.01 | 8.98 | – | STFT CNN | Yes | Yes | 5 | STFT, $N_{\text{fft}} = 25, H = 20$ |
| 74.37 | 64.86 | 4.29 | – | CNN + Transformer | Yes | Yes | 1 | |
| – | 51.68 | 3.33 | – | CNN + Transformer | Yes | Yes | 1 | Ternary PanPhon, |
| 75.22 | 63.82 | 0.98 | – | CNN + Transformer | Yes | Yes | 1 | |
| – | 57.21 | 0.88 | – | CNN + Transformer | Yes | Yes | 1 | Binary PanPhon, |
| – | – | 0.06 | – | CNN | Yes | Yes | 1 | w/ Distribution Mapper |
| – | – | 0.12 | – | CNN + Transformer | Yes | Yes | 1 | w/ Distribution Mapper |

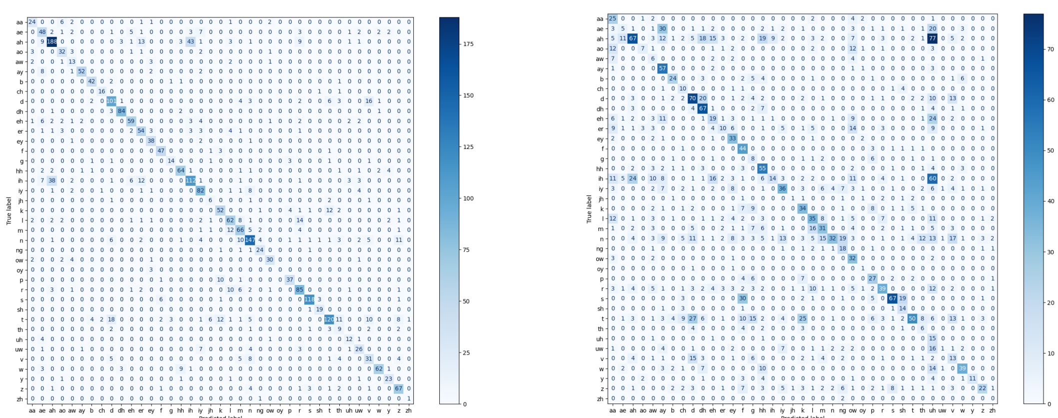

Figure 4: Confusion matrices for the 2nd configuration in Table 5 on validation (left) and leaderboard (right) data.