CTC vs. Seq2Seq Learning: Two Approaches for Automatic Music transcription

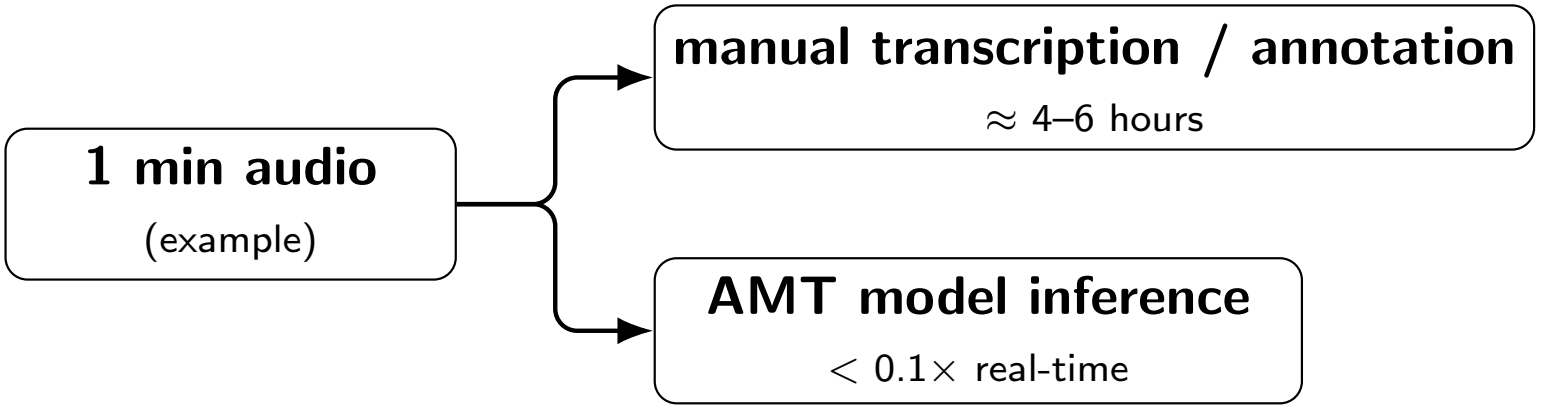
Livia Kastrati Phillip Olshausen

HTW Berlin

January 22, 2026

Overview

Why Automatic Music Transcription (AMT)?



scale gap
⇒ AMT

cost of labels & transcription → bottleneck

Relevance?

- music education / feedback
- production: audio → editable MIDI
- MIR: symbolic search / indexing
- datasets: faster annotation

The Challenge of Music Transcription

continuous audio

$$\Downarrow$$

discrete musical events

Audio domain

- dense signal
- overlapping frequencies
- no explicit boundaries

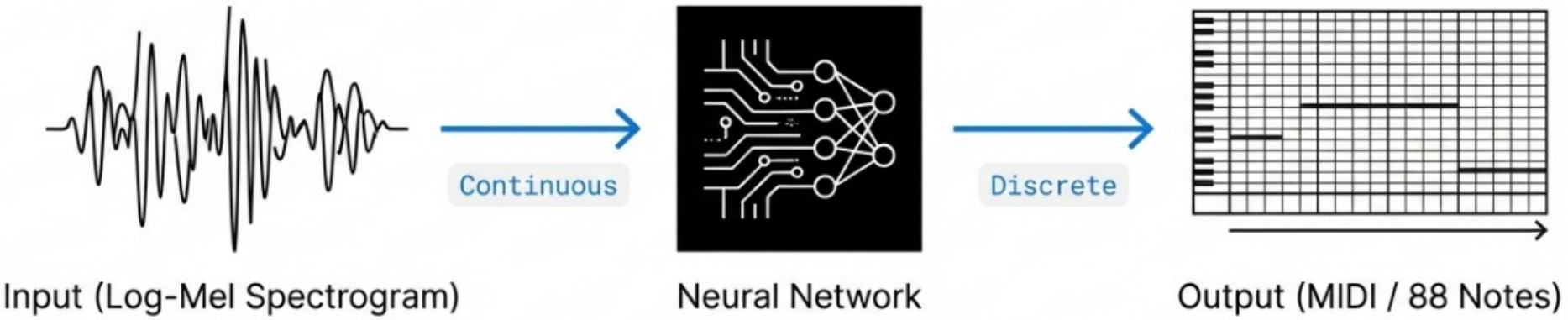
Symbolic domain

- note on / off
- pitch + timing
- long-range structure

Problem type

supervised · structured prediction

Audio-to-Note Transcription



- **Input:** audio, represented as a time–frequency feature
- **Model:** learns a mapping from acoustic evidence $\rightarrow$ *musicalevents*.

- **Output:** discrete symbolic representation

  In our setting: fixed encoder architecture fixed and varied input representation.

CTC vs Seq2Seq: What Is Being Predicted?

CTC: "what's sounding now?" vs Seq2Seq: "what happens next?"

CTC

- **Prediction unit:** frame token k_t (pitch ID or BLANK)
- **Vocabulary:** MIDI pitches + blank($\varnothing$)
- **Output:** fixed length T (one distribution per frame)
- **Decode:** collapse repeats $\rightarrow$ remove blanks

Example:
[60, 60, 61, $\varnothing$, 60, $\varnothing$] $\rightarrow$ [60, 61, 60]

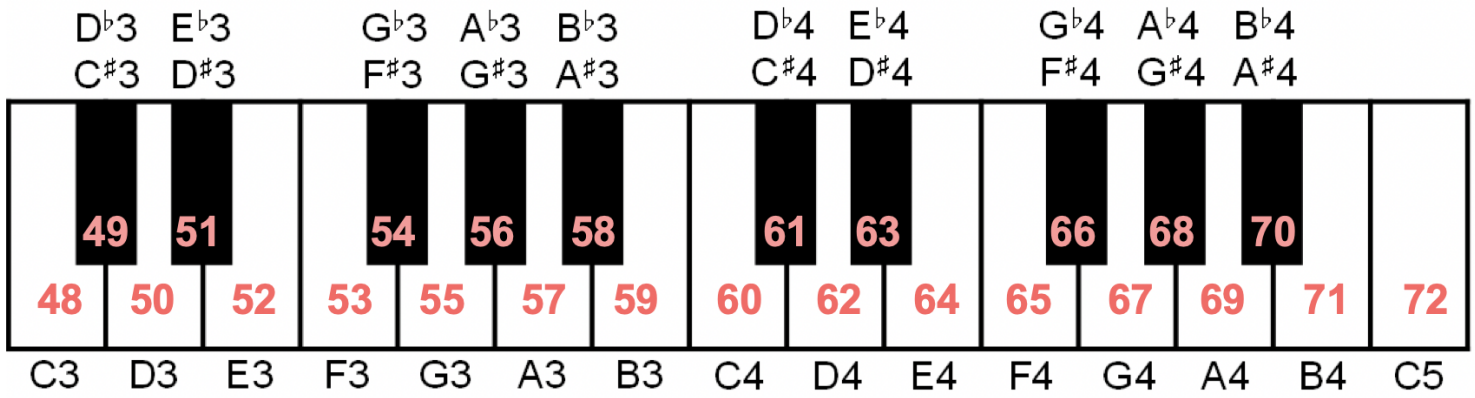
Seq2Seq

- **Prediction unit:** next event token s_t
- **Vocabulary:** (`NOTE_ON/OFF`, `TIME_SHIFT`, `EOS`)
- **Output:** variable-length event sequence
- **Decode:** autoregressive until `EOS`

Example:
`NOTE_ON(60)` `TIME_SHIFT(k)` `NOTE_OFF(60)` ...

From sound to symbols: why we use MIDI pitches

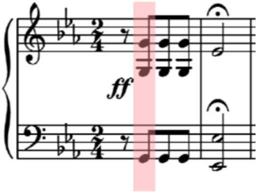
MIDI note numbers
(MIDI pitches)



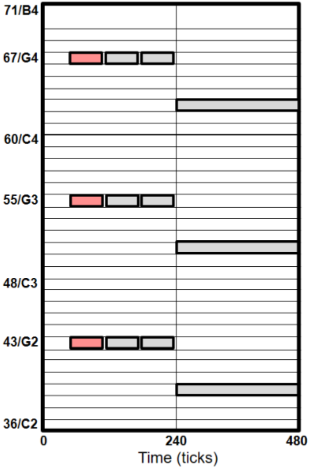
- For learning and evaluation, we need **discrete targets**.

MIDI: Musical Instrument Digital interface

Figure 1.13 from
[Müller, FMP, Springer 2015]



Time (Ticks)	Message	Channel	Note Number	Velocity
60	NOTE ON	1	67	100
0	NOTE ON	1	55	100
0	NOTE ON	2	43	100
55	NOTE OFF	1	67	0
0	NOTE OFF	1	55	0
0	NOTE OFF	2	43	0
5	NOTE ON	1	67	100
0	NOTE ON	1	55	100
0	NOTE ON	2	43	100
55	NOTE OFF	1	67	0
0	NOTE OFF	1	55	0
0	NOTE OFF	2	43	0
5	NOTE ON	1	67	100
0	NOTE ON	1	55	100
0	NOTE ON	2	43	100
55	NOTE OFF	1	67	0
0	NOTE OFF	1	55	0
0	NOTE OFF	2	43	0
5	NOTE ON	1	63	100
0	NOTE ON	2	51	100
0	NOTE ON	2	39	100
240	NOTE OFF	1	63	0
0	NOTE OFF	2	51	0
0	NOTE OFF	2	39	0



Dataset

MusicNet (Thickstun et al., 2017)

- **Scale:** 330 freely-licensed classical music recordings.
- **Annotations:** over **1.29M** aligned note events with *start/end time*, *instrument*, *pitch (MIDI note)*, and *metrical position*
- **Coverage:** recordings span 10 composers and 11 instruments; average duration $\approx$ 6 minutes per recording
- **Imbalance:** skewed toward Beethoven and solo piano

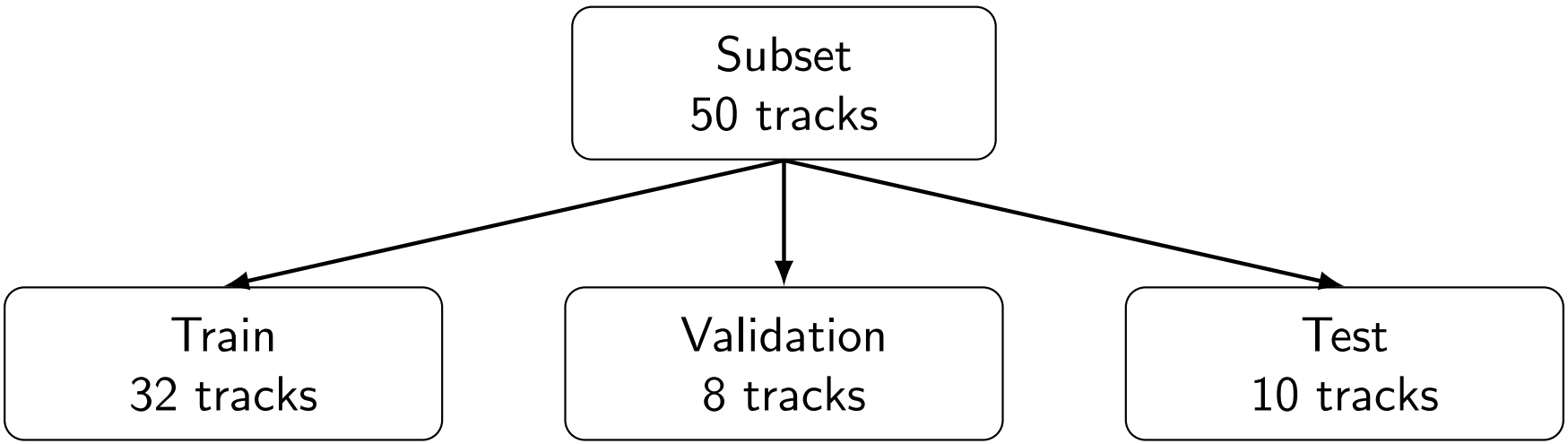
Subset

- Motivation: disk space constraints.
- Selection of **40** recordings with the most labeled note events.
- Final split: **50 tracks total** (test set unchanged: **10** recordings).

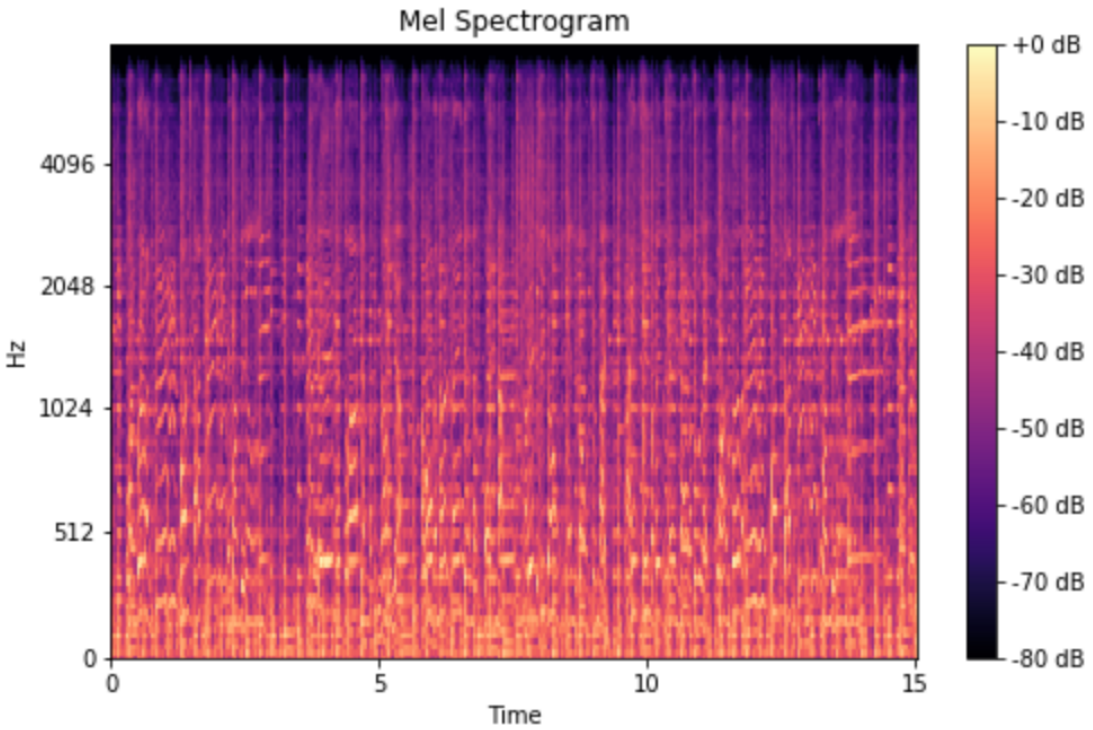
Train/Validation/Test Split

Split rule:

- Test set unchanged.
- Validation is sampled from train pool **by musical work**.
- Work key: `(composer, composition)`



Preprocessing



log-mel spectrogram (time $\times$ frequency)

Input transformation

- waveform $\rightarrow$ STFT
- mel filterbank
- log magnitude

Why time-frequency feature

- compact representation
- perceptually motivated
- stable for training

Encoder Overview (CNN + BiLSTM)

- Input: time-frequency feature

$$X \in \mathbb{R}^{B \times T \times F}$$

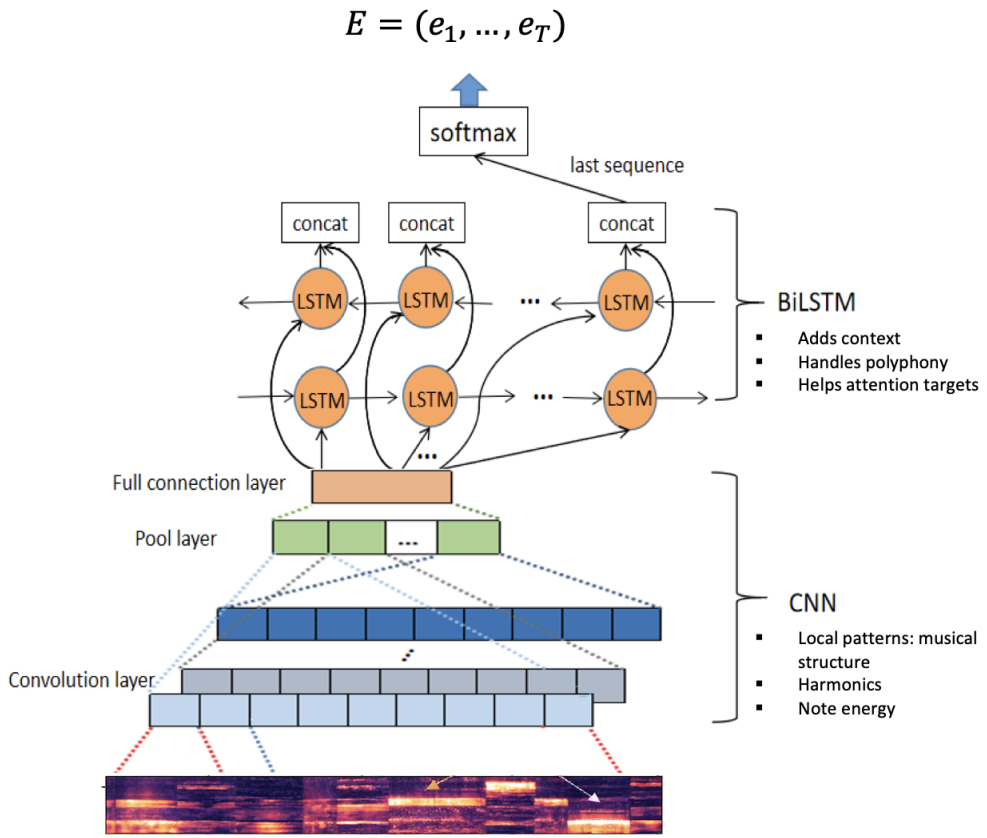
- CNN feature extractor: local time–frequency patterns
- Reshape/permute to sequence:

$$Z \in \mathbb{R}^{B \times T' \times d}$$

- BiLSTM temporal modeling:

$$E = (e_1, \ldots, e_{T'}), \quad e_i \in \mathbb{R}^{d_{\text{enc}}}$$

- Output: encoder states E + lengths/mask



Generic CNN-BiLSTM pipeline illustration

(adapted from: Omarov et al., 2023.)

Encoder: BiLSTM States & Padding Mask

BiLSTM temporal encoder

$$\overrightarrow{H} = \mathrm{LSTM}_f(Z), \qquad \overleftarrow{H} = \mathrm{LSTM}_b(Z)$$

$$E_t = \left[\overrightarrow{H}_t; \overleftarrow{H}_t\right] \in \mathbb{R}^{d_{\mathrm{enc}}}, \qquad E \in \mathbb{R}^{B \times T' \times d_{\mathrm{enc}}}$$

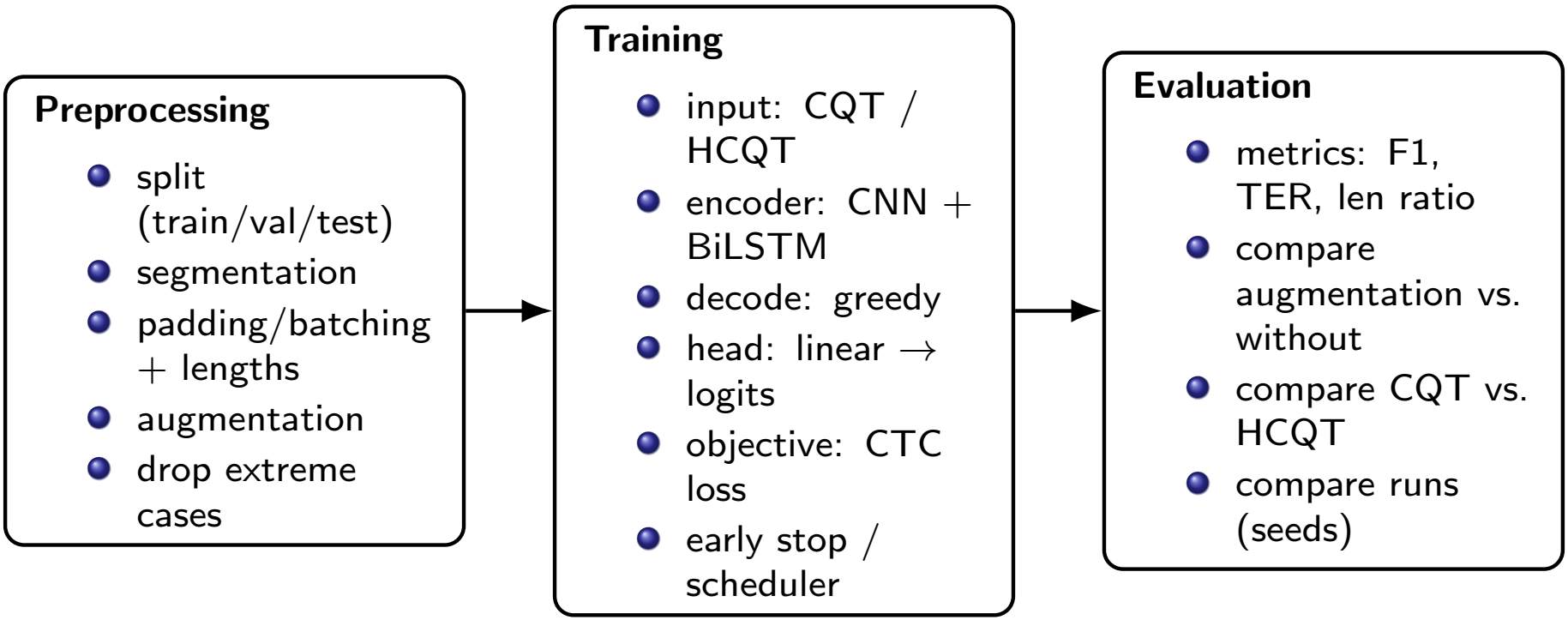
Lengths & mask (for attention)

$$\ell \in \mathbb{N}^B, \qquad M_{b,t} = \mathbb{1}[t < \ell_b] \in \{0, 1\}$$

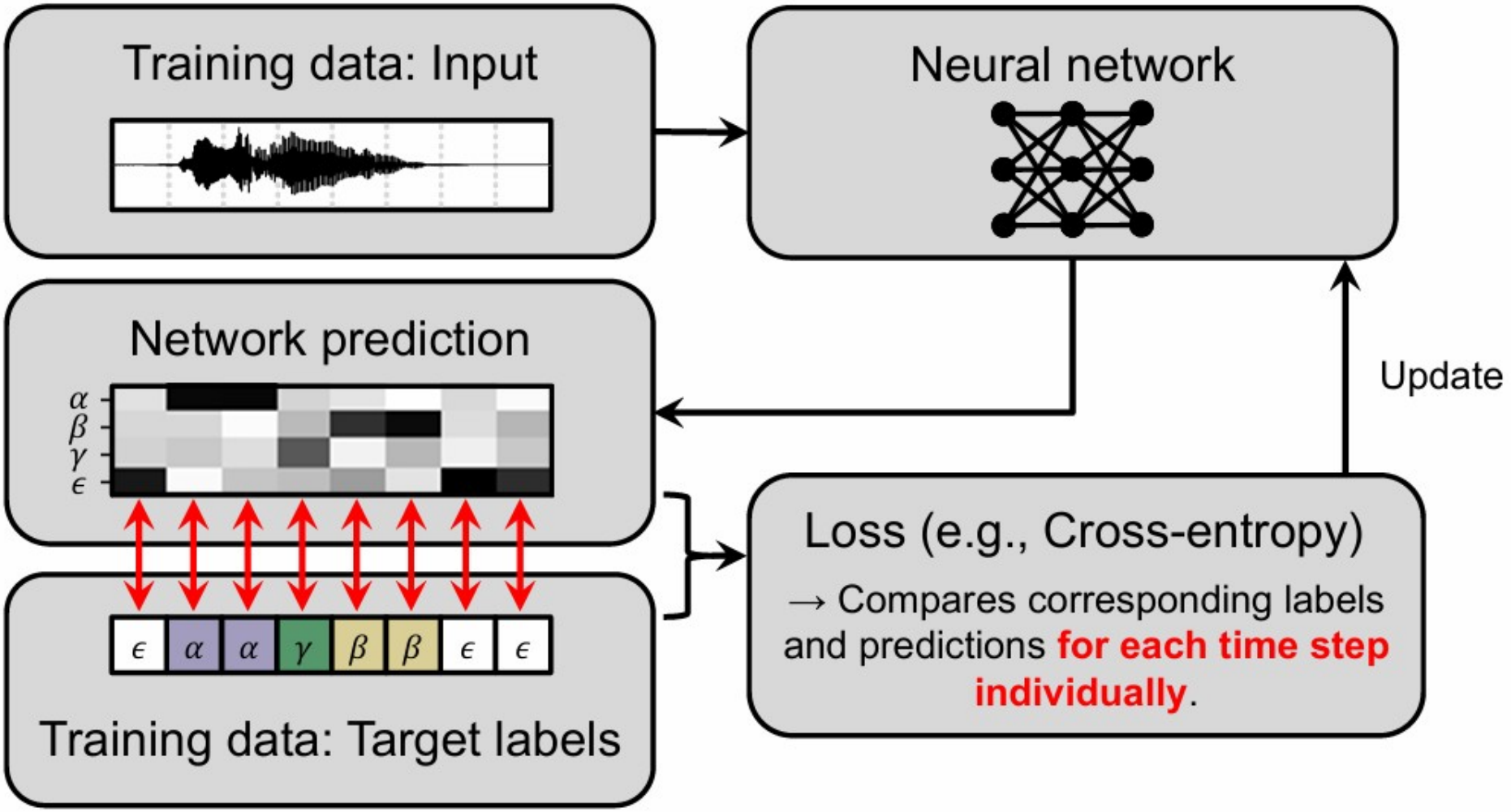
output to decoder: (E, ℓ) / (E, M)

- Z : CNN feature sequence
- E : encoder state sequence
- d_{enc} : BiLSTM state dim
- ℓ : valid lengths
- M : padding mask

CTC Overview

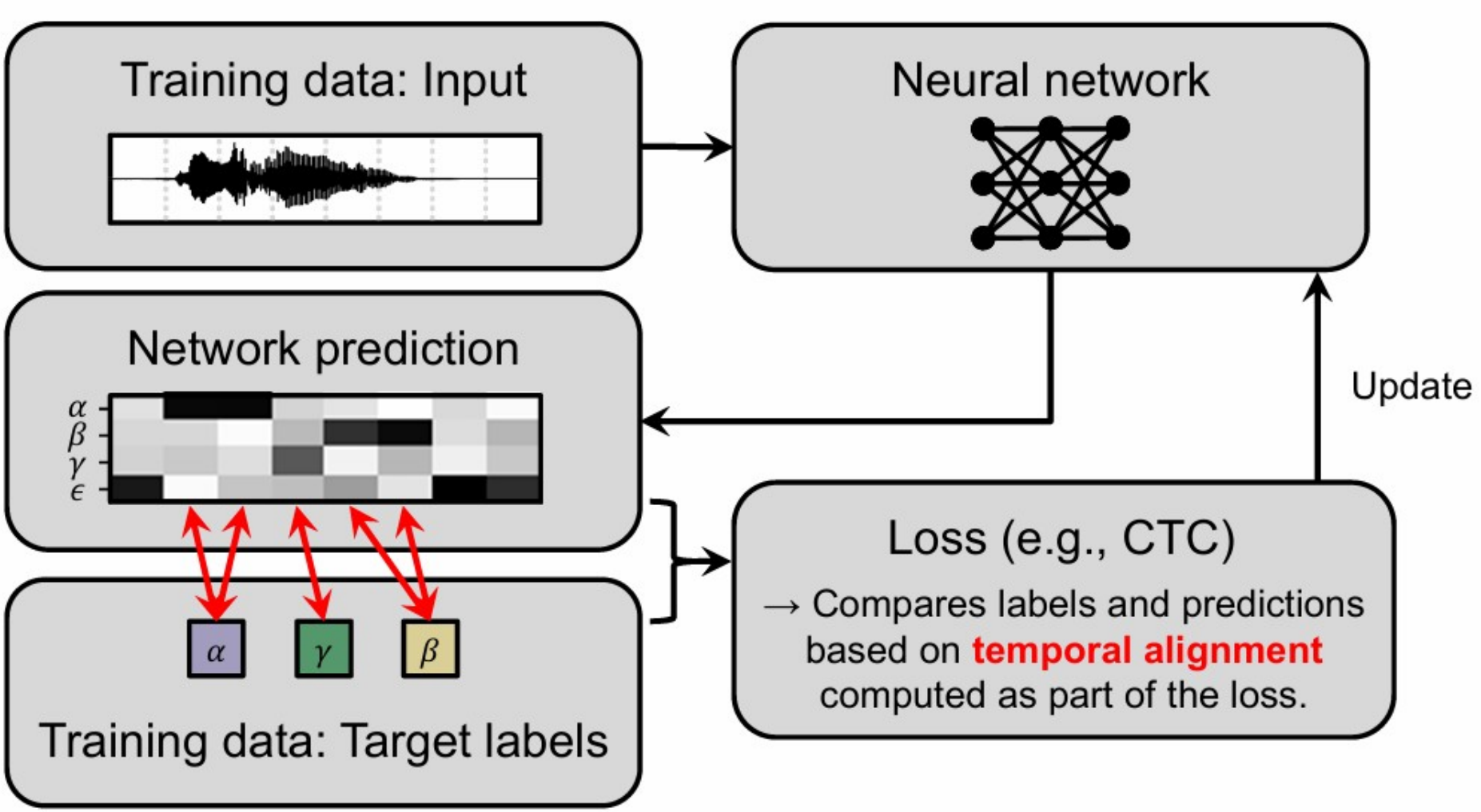


Strongly Aligned Training Data



Source: Zalkow & Müller (2021), *CTC loss* lecture notes.

Weakly Aligned Training Data



Source: Zalkow & Müller (2021), *CTC loss* lecture notes.

Output Representation

Pitch range:

MIDI pitches in the piano range: $\mathcal{P} = \{21, 22, \ldots, 108\}, \quad |\mathcal{P}| = 88.$

Token Mapping:

Each pitch is represented by a token ID. The pitch-token alphabet is:

$$\mathcal{A} = \{0, 1, \ldots, 87\}, \qquad |\mathcal{A}| = 88.$$

ID = note - 21 $\in \{0, \ldots, 87\}$

MIDI pitch p	21	22	$\ldots$	60	64	108
Token id $p - 21$	0	1	$\ldots$	39	43	87

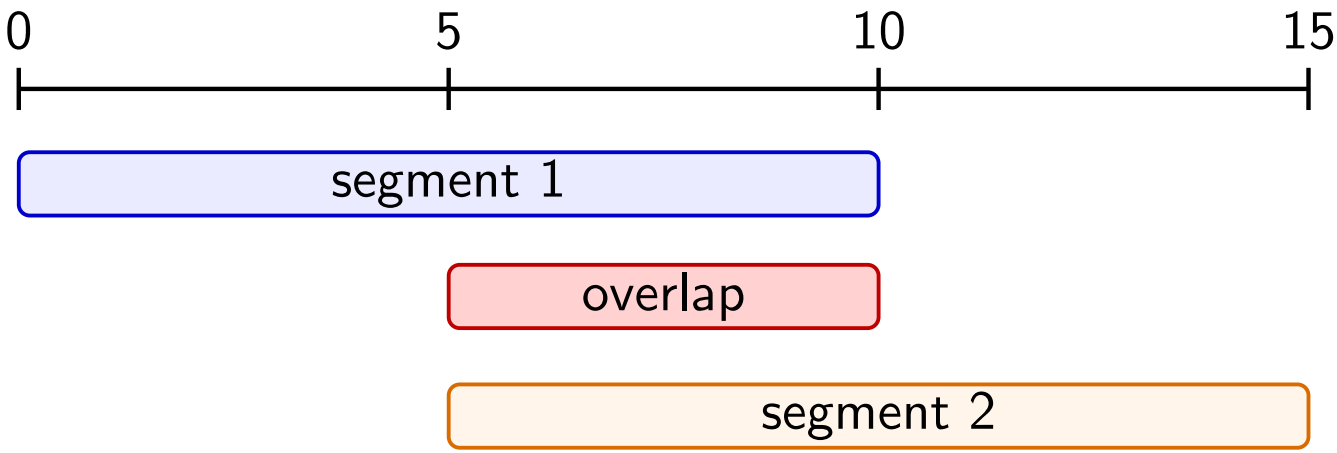
Special tokens:

$$\text{PAD} = 88, \quad \text{BLANK} = 89, \quad \text{SOS} = 90, \quad \text{EOS} = 91$$

Segmentation

Segment Length (L): the duration of one audio chunk processed at a time.
Hop Size (H): how far the window moves forward to create the next segment.

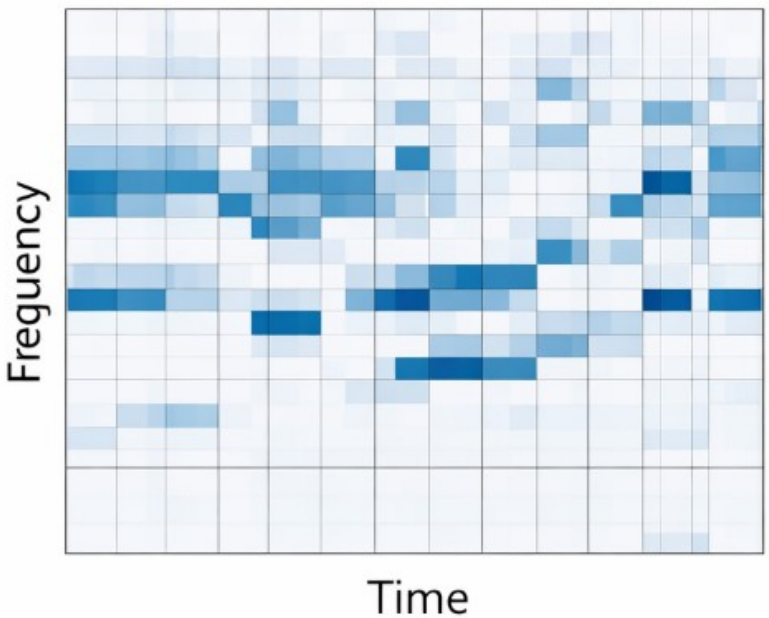
- $L = \mathbf{10}$ seconds per segment.
- $H = \mathbf{5}$ seconds between segments. (50% overlap)
- **Skip the final incomplete tail window** to keep consistent segment shapes.



CQT vs. HCQT

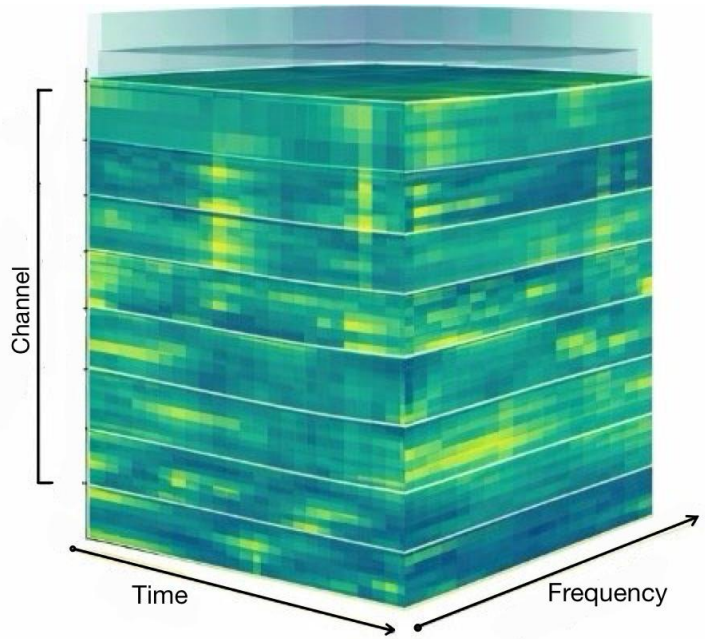
Constant-Q Transform (CQT)

- 2D tensor: **Time** $\times$ **Frequency**
- Log-frequency axis (music-friendly spacing)
- Our setup: **252 bins** (**36 bins/octave**)



Harmonic CQT (HCQT)

- 3D tensor: **Time** $\times$ **Channel** $\times$ **Frequency**
- Channels stack harmonic views of the spectrum
- Our setup: harmonics $h \in \{0.5, 1, 2, 3, 4\}$ (5 channels)



Stabilizing Training: Removing Extreme Segments

- Some segments contain **very dense chords**.
- We inspect token-length distributions for **train/val/test**.
- **Train-only filter:** keep segments with $n\_tokens \leq 180$ (removes rare extreme cases).
- The 90th percentile is ≈ 175 tokens so we cut only the long tail.
- Validation and test remain unchanged.

Split	#before	#now	Change
Train	698	635	−63
Val	186	186	0
Test	280	280	0

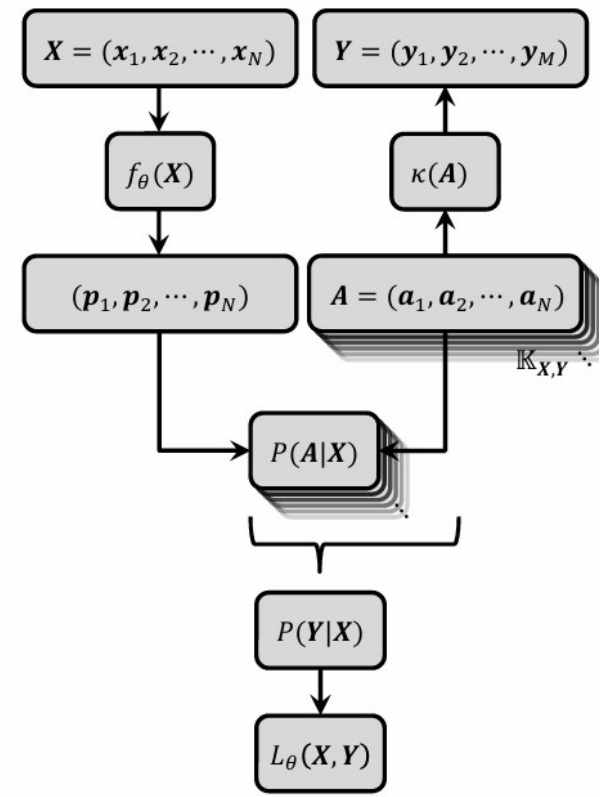
Augmentation

Augmentation Type	**Changes**
Small pitch / tuning shift	up to ± 2 bins
Transposition-like shift	up to ± 5 bins
Additive noise	$\sigma \approx 0.02$, applied with $p = 0.5$
Time masking	up to 40 frames, 2 masks
Frequency masking	up to 24 bins, 2 masks

Applied **only during training**. Validation and test remain unchanged.
Augmentation motivated by invariance assumptions for AMT (Thickstun et al., 2018).

CTC Model

- **Input sequence X:** a 10s audio segment turned into a time–frequency feature
  - CQT or HCQT
- **Network $f_\theta(X)$: CNN + BiLSTM** encoder
  - CNN learns local time–frequency patterns
  - BiLSTM models temporal context across frames
- **Per-frame outputs $(p_1, \ldots, p_N)$:** a softmax distribution *at every frame*
  - vocabulary $A' = \{0, \ldots, 87\} \cup \{\texttt{BLANK}\}$ (88 pitches + blank)
- **Target Y:** the pitch-token sequence for this segment (order only; no durations)

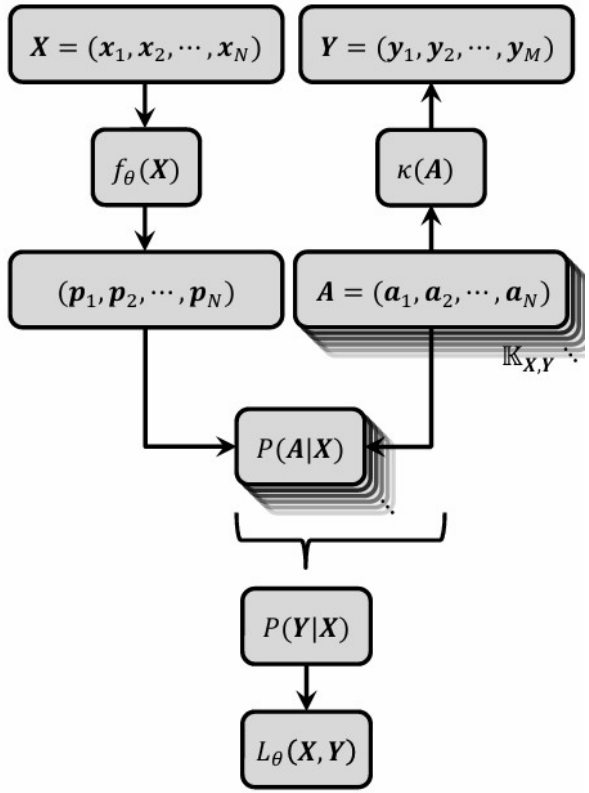


Source: Zalkow & Müller (2021), *CTC loss* lecture notes.

CTC Model

- The model outputs a **token distribution at each frame**.
- A **path / alignment** $A = (a_1, \ldots, a_N)$ picks one token per frame (a pitch ID **or** `BLANK`)
- A **collapse function** $\kappa(\cdot)$ turns a path into a label sequence:
  - **merge repeats**
  - **remove blanks**
- Key point: **many different paths** A collapse to the **same** target Y.
- CTC trains by **maximizing the conditional probability** of the correct label sequence:

$$\max_{\theta} P(Y \mid X) = \max_{\theta} \sum_{A:\, \kappa(A)=Y} P(A \mid X).$$



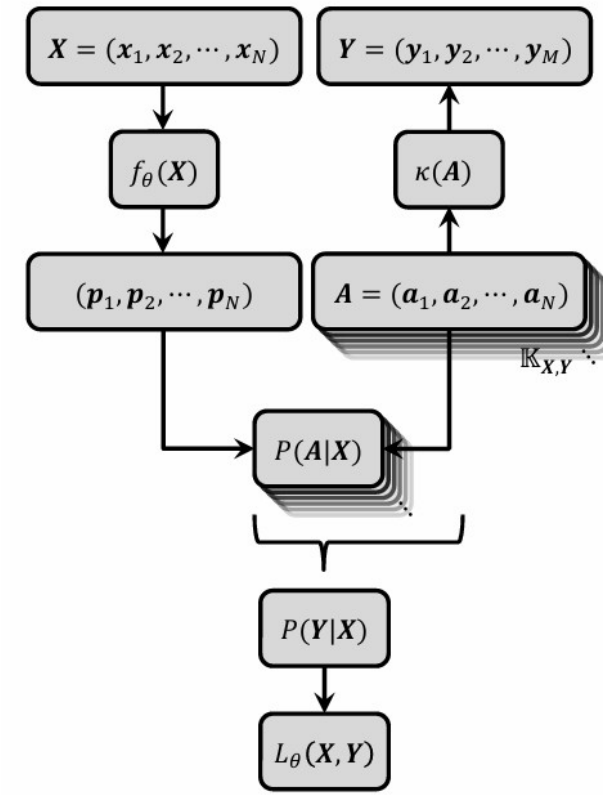
Source: Zalkow & Müller (2021), *CTC loss* lecture notes.

CTC Model

- **Training loss:**

$$\mathcal{L}_{\text{CTC}}(X, Y) = -\log P(Y|X)$$

- Intuition: the encoder is rewarded if it assigns high probability to **any** frame-wise path that collapses to the correct pitch sequence.
- **Decoding:**
  - **Greedy:** pick the most likely token per frame $\rightarrow$ collapse repeats $\rightarrow$ drop blanks
- **Training:** Optimized with Adam; early stopping on validation loss (ReduceLROnPlateau).



Source: Zalkow & Müller (2021), *CTC loss* lecture notes.

Comparison Table

- Factors: **Feature** $\in \{\text{CQT}, \text{HCQT}\}$, **Augmentation** $\in \{\text{Off}, \text{On}\}$ (all else fixed).
- Key metrics: **F1**, **TER**, **len_ratio** $= |\hat{Y}|/|Y|$.

Feature	Aug	test_F1	test_TER	len_ratio
CQT	Off	0.695	0.531	0.644
HCQT	Off	0.724	0.505	0.743
CQT	On	0.000	1.000	0.000
HCQT	On	0.021	0.990	0.012

Headline: Without augmentation, **HCQT improves F1 and TER** vs CQT. With augmentation, both models collapse.

Takeaway

- **Representation effect (no aug):** HCQT $>$ CQT in F1 and TER.
- **Augmentation effect:** current augmentation causes severe under-generation (blank collapse), dominating performance.

Best configuration	**test_F1**	**test_TER**	**len_ratio**
HCQT + Aug Off (seed=42)	0.724	0.505	0.743

Experiment: CQT vs. HCQT (3 seeds)

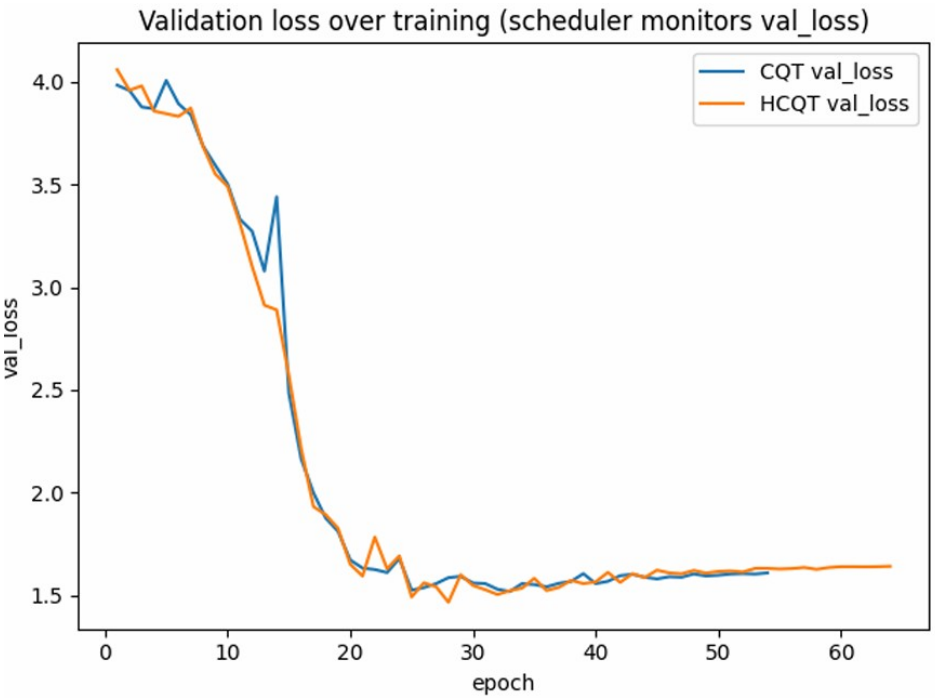
- Goal: isolate the effect of the **input representation** on CTC transcription.
- Compared features:
  - **CQT** (1 channel)
  - **HCQT** (5 harmonic channels)
- Augmentation: **OFF**
- **New Setup:**
  - **Early stopping on validation F1**
  - min_epochs $= 40$
  - patience $= 15$,
  - min_delta_F1 $= 0.003$
- Protocol: same data split, same model, same training procedure; **3 random seeds**.
- Reporting: mean $\pm$ std across seeds on the **test set**.

Experiment Results

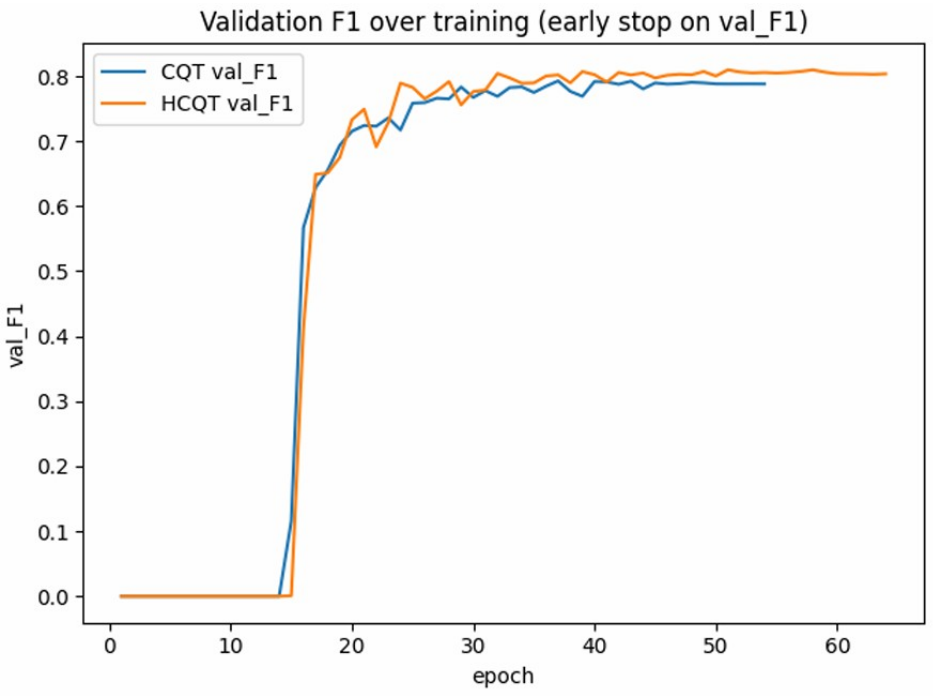
Feature	test_F1	test_TER	test_len_ratio	best epoch
CQT	0.7378 ± 0.0000	0.5129 ± 0.0000	0.8489 ± 0.0000	37
HCQT	**0.7471** ± 0.0000	**0.4943** ± 0.0000	0.8423 ± 0.0000	49

Note: the displayed std rounds to 0.0000 (variance across seeds is negligible at this precision).

Training Plots

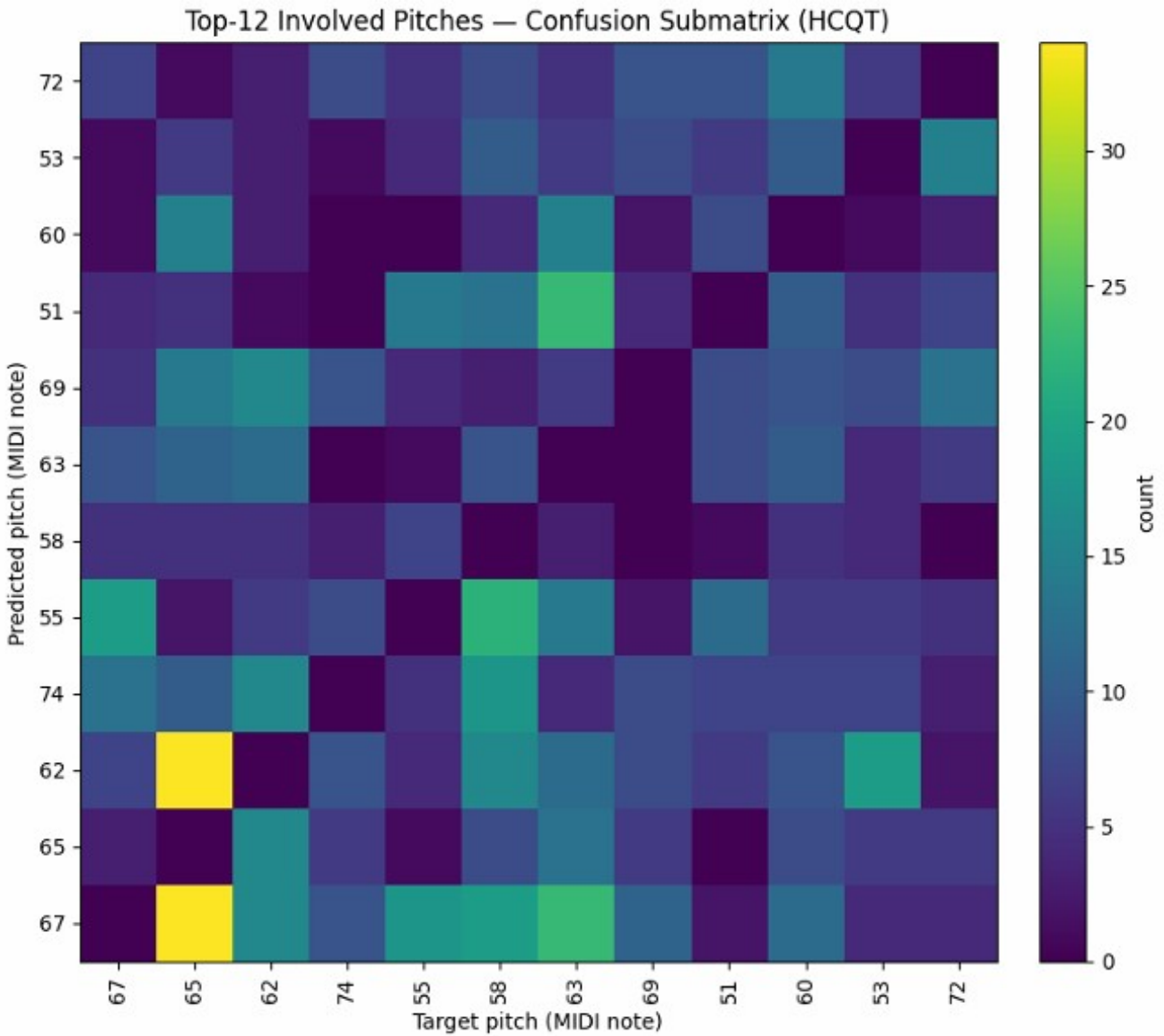


Loss (train vs val)



Validation F1

HCQT Confusion Matrix



Key takeaways

- **Overall performance (seed=42, no augmentation):**
  - **HCQT slightly improves** token-F1 and TER, but the margin is small.
- **Important metric caveat (explains "high F1 but looks wrong"):**
  - Reported F1 is *bag-of-tokens* overlap (counts TP/FP/FN), not time-aligned accuracy.
  - Therefore, F1 can be high even when the predicted pitch sequence is mis-ordered or mis-timed.
- **Confusion matrices / tables are error-focused:**
  - "Top substitutions" explicitly excludes correct matches (diagonal).
  - The displayed subtable emphasizes *substitution errors*, not overall correctness.
- **Main failure mode (CTC + greedy):**
  - len_ratio ≈ 0.84 indicates under-production (blank dominance / collapse).
  - Worst segments show high TER and low token-F1 concentrated in dense/polyphonic passages.

Why Seq2Seq

audio $\rightarrow$ symbolic events

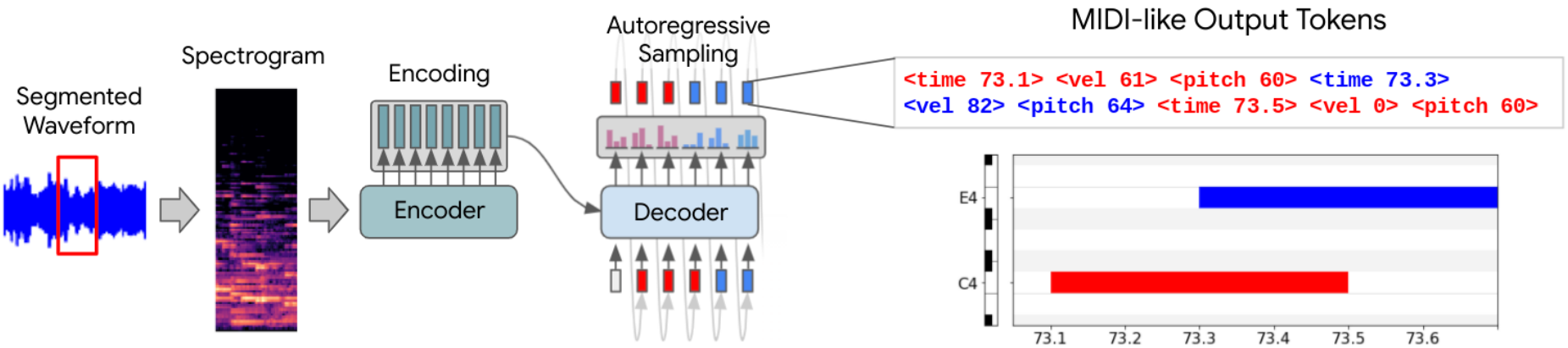
Challenges

- polyphony
- overlapping notes
- variable timing
- long-range structure

Why Seq2Seq + Attention

- variable-length output
- explicit alignment
- event-level modeling
- global musical context

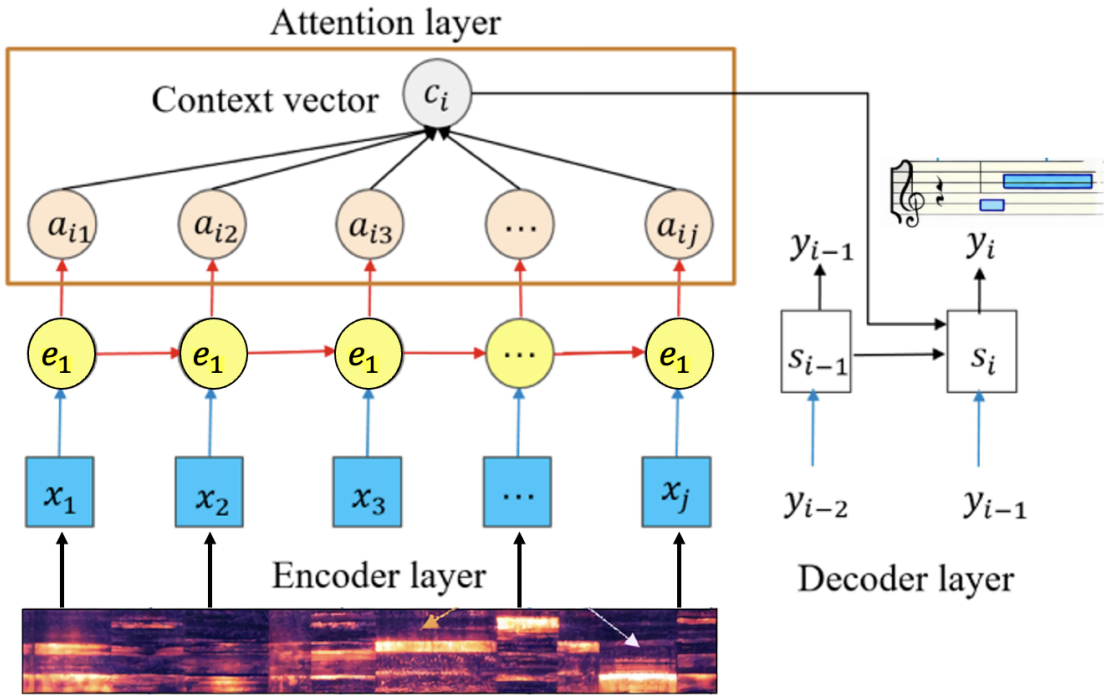
Seq2Seq: Architecture



- **Origin:** Neural machine translation (Sutskever et al., 2014)
- **Purpose:** variable-length input $\rightarrow$ variable-length output

Attention Intuition (Additive / Bahdanau)

- **Decoder query** $s_t = h_t$: *"what do I need now?"*
- **Encoder states** $E = \{e_i\}$: *"where is info in audio?"*
- Attention learns **alignment**: output events $\leftrightarrow$ relevant encoder times



Graphic adapted from: (Yang et al., 2021, Fig. 1).

Additive attention: weights $\alpha_{t,i}$ form context c_t.

$$\alpha_{t,i} = \mathrm{softmax}(s_{t,i}), \quad c_t = \sum_i \alpha_{t,i} e_i$$

Additive Attention (Bahdanau): Main Formula

$$s_{t,i} = v^\top \tanh\big(W_h h_t + W_e e_i + b\big), \qquad \alpha_{t,i} = \mathrm{softmax}_i(s_{t,i})$$

$$c_t = \sum_{i=1}^{T'} \alpha_{t,i}\, e_i$$

Variables

- t = decoder step, i = encoder index, T' = encoder length
- $h_t \in \mathbb{R}^{d_{\mathrm{dec}}}$ = decoder hidden (query)
- $e_i \in \mathbb{R}^{d_{\mathrm{enc}}}$ = encoder state (keys/values)
- $s_{t,i}$ = alignment score, $\alpha_{t,i}$ = attention weight ($\sum_i \alpha_{t,i} = 1$)
- $c_t \in \mathbb{R}^{d_{\mathrm{enc}}}$ = context vector

Parameters $W_h \in \mathbb{R}^{d_a \times d_{\mathrm{dec}}}$, $W_e \in \mathbb{R}^{d_a \times d_{\mathrm{enc}}}$, $v \in \mathbb{R}^{d_a}$, $b \in \mathbb{R}^{d_a}$

Decoder Loop (Architecture): Input Feeding + Attention

- **Input feeding:**

$$x_t = [\mathrm{emb}(y_{t-1});\ c_{t-1}]$$

- **Recurrent update:**

$$s_t = \mathrm{LSTM}(x_t, h_{t-1})$$

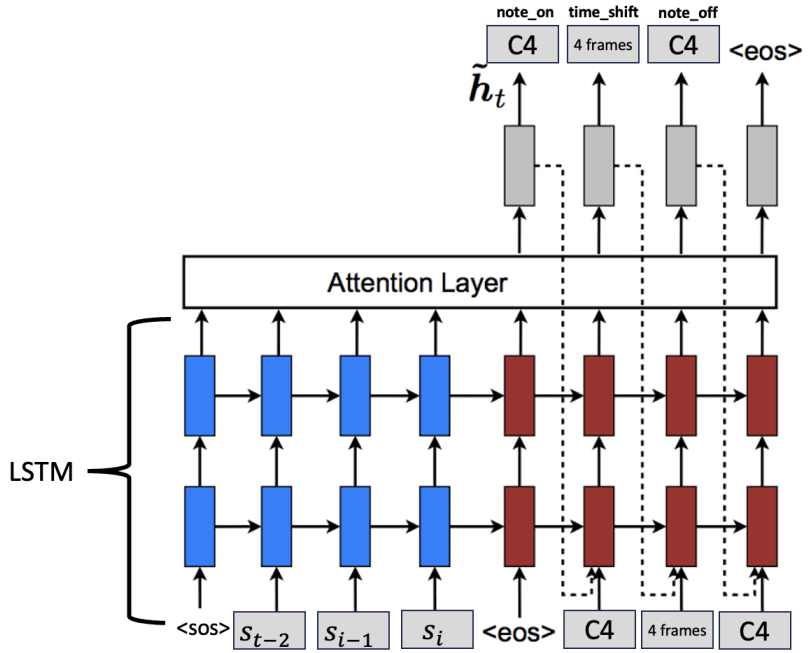
- **Attention over encoder states $E = \{e_i\}$:**

$$c_t = \mathrm{Attn}(h_t, E)$$

- **Predict next token:**

$$\ell_t = W_o[h_t; c_t] \quad \Rightarrow \quad y_t$$

Training: teacher forcing (y_{t-1} from target)



Input-feeding idea: feed attention-derived vector to next step (Luong et al., 2015, Fig. 4).

Training Objective (Weighted Next-Token Cross-Entropy)

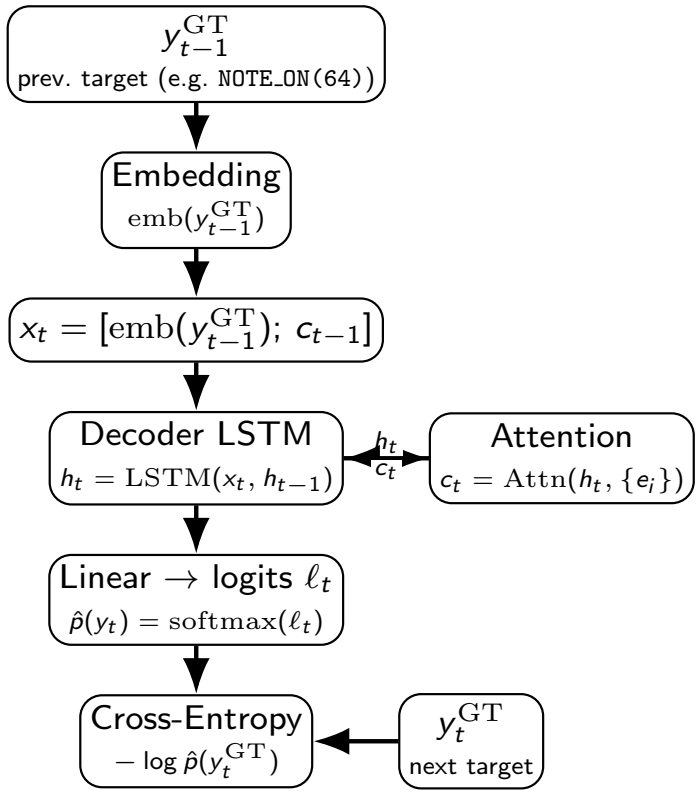
$$\mathcal{L}(\theta) = \sum_{t=1}^{U} \underbrace{- w_{s_t} \log p_\theta(s_t \mid s_{<t}, X)}_{\text{Weighted NLL at step } t}$$

Variables

- $X \in \mathbb{R}^{T \times M}$: log-mel segment (frames $\times$ mel bins), M=128.
- $s_{1:U}$: target event-token sequence; U = number of tokens in the segment.
- $s_{<t}$: previous tokens (decoder context).
- $p_\theta(\cdot)$: decoder softmax distribution over token vocabulary $\mathcal{V}$.
- w_{s_t}: token weight (class imbalance: TIME_SHIFT / NOTE_ON/OFF / EOS).
- θ: all model parameters (encoder + attention decoder).

Minimize negative log-likelihood of the correct next event token, with token reweighting.

Teacher Forcing (Training)



Teacher forcing: previous token is **ground truth** y_{t-1}^{GT} (not $\hat{y}_{t-1}$).

Fallbacks / Safeguards

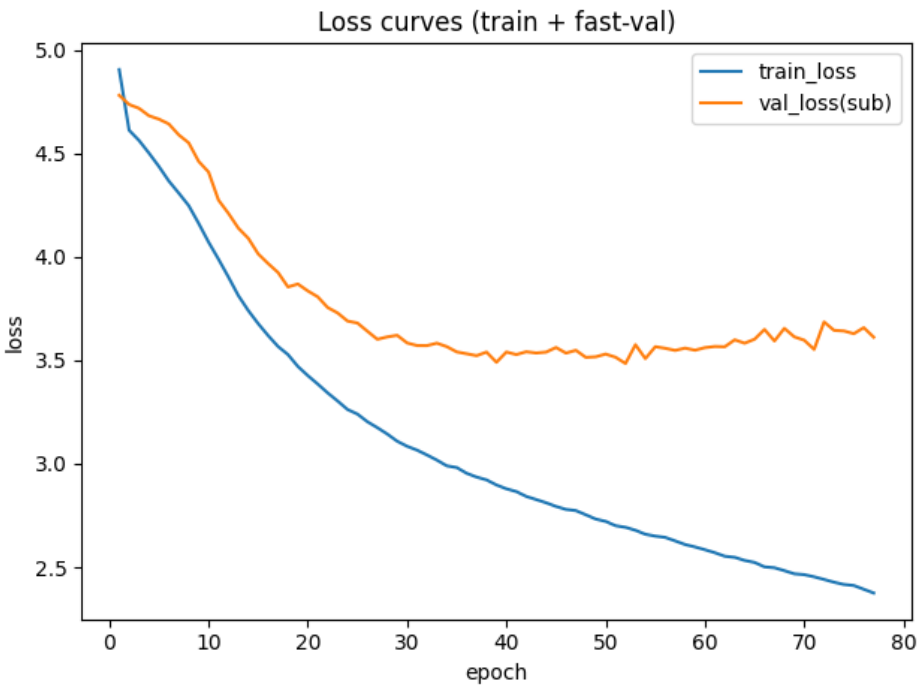
- decoding stop rules
  - EOS
  - length cap: $t \leq U_{\max}$
  - repeated EOS / degenerate loops
- attention masking
  - pad $\rightarrow$ fill -10^4 before softmax
  - energies in fp32 (AMP-safe)
- training stability
  - gradient clipping
  - NaN/Inf guard (skip step)
  - gradient accumulation (GPU fit)
- loss masking
  - ignore PAD targets

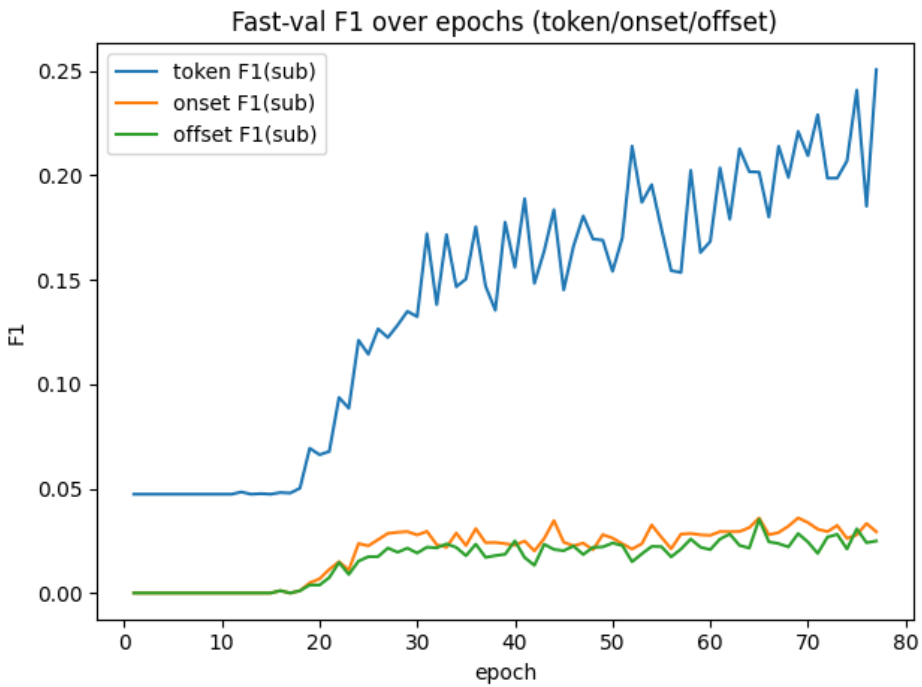
Iterative Updates

Changes driven by metrics

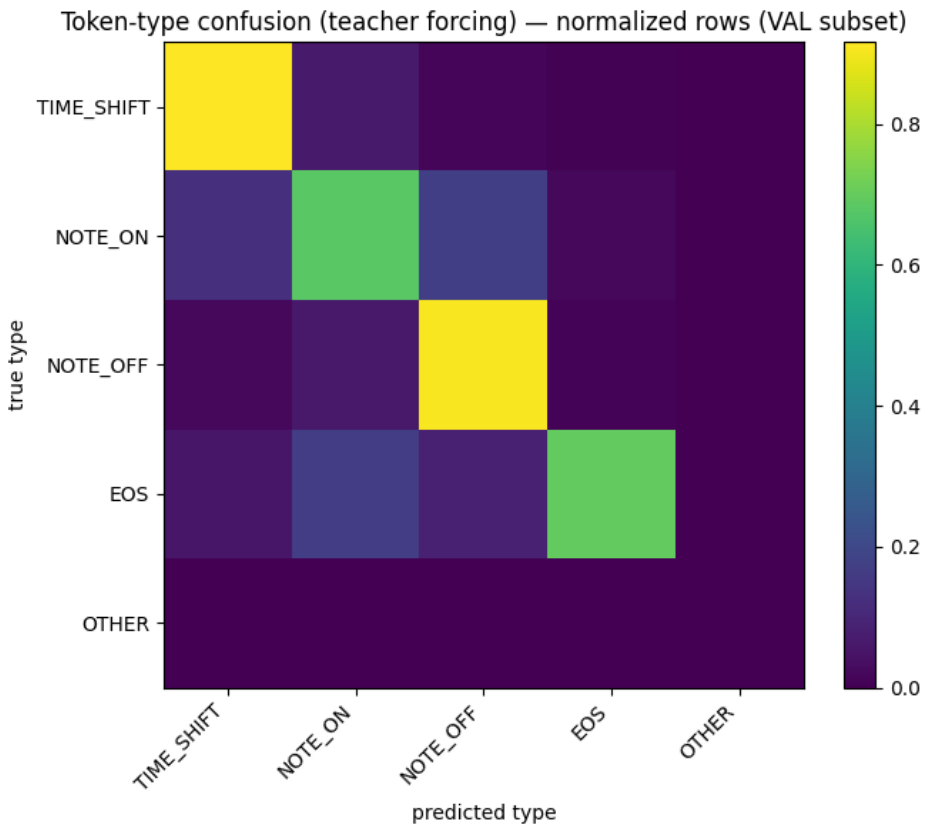
- class weights: TIME_SHIFT / EOS emphasis
- eval: incremental token-F1 (no ragged concat)
- AMP-safe masking: fp32 energies + -10^4
- decoding: cap $U_{\max}$ + stop rules
- grad accumulation + clip (stable convergence)

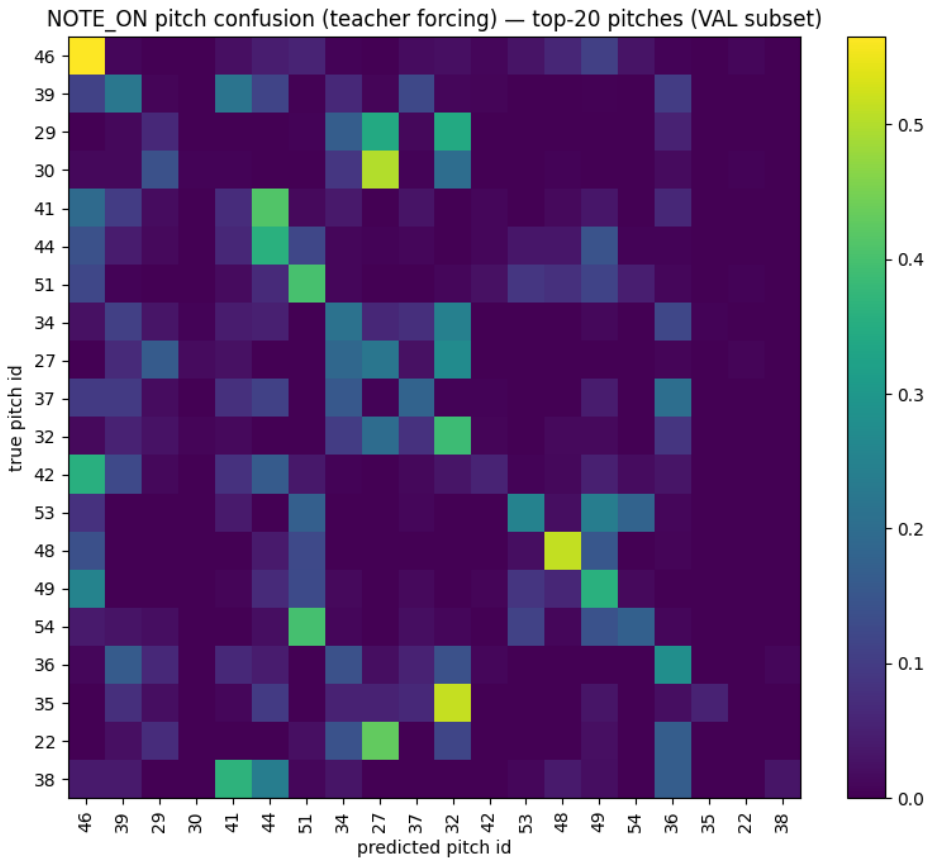
Results


Loss curves (train + fast-val)
train_loss
val_loss(sub)
loss
epoch
5.0
4.5
4.0
3.5
3.0
2.5
0
10
20
30
40
50
60
70
80

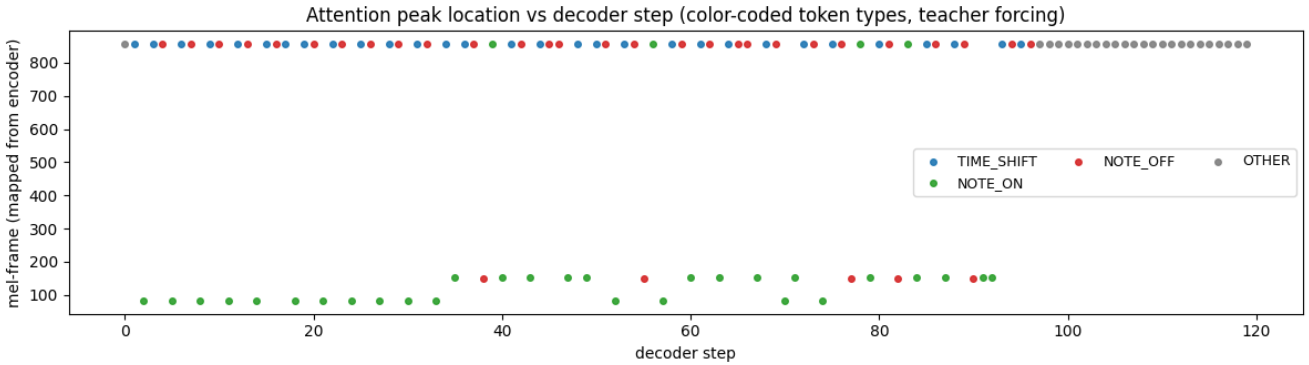

Fast-val F1 over epochs (token/onset/offset)
token F1(sub)
onset F1(sub)
offset F1(sub)
F1
epoch
0.25
0.20
0.15
0.10
0.05
0.00
0
10
20
30
40
50
60
70
80

Confusion Matrices (Validation, Teacher Forcing)



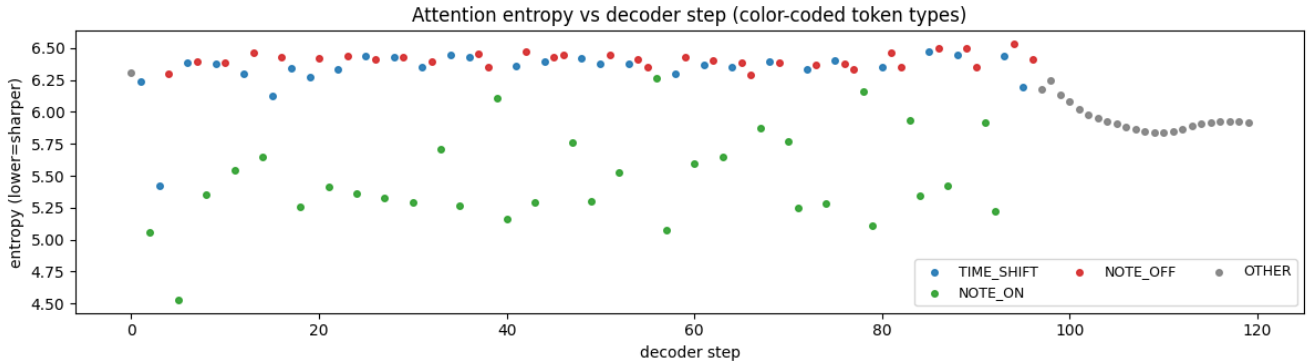


Attention Diagnostics (Teacher Forcing)



Peak plot

- token step t vs encoder frame index
- color: token type



Entropy plot

- attention sharpness over decoder steps
- color: token type

Takeaway

- Under identical training conditions (3 seeds), is learning but lacks information for good results.
- Token-prediction works well but pitch-prediction struggles.
- lower metrics but actually predicts pitches better
- attention still in very early stages
- Next: train on larger dataset.

References

- Thickstun, J., Harchaoui, Z., & Kakade, S. M. (2017). *Learning features of music from scratch*. ICLR.
- Thickstun, J., Harchaoui, Z., Foster, D., & Kakade, S. M. (2017). *Invariances and data augmentation for supervised music transcription*.
- Zalkow, F., & Müller, M. (2021). *CTC-based learning of chroma features for score–audio music retrieval*. IEEE/ACM TASLP.
- Zalkow, F., & Müller, M. (2021). *Connectionist Temporal Classification (CTC) loss*. Lecture notes, IAL Erlangen.
- Sutskever, I., Vinyals, O., & Le, Q. V. (2014). *Sequence to Sequence Learning with Neural Networks*. NeurIPS.
- Bahdanau, D., Cho, K., & Bengio, Y. (2015). *Neural Machine Translation by Jointly Learning to Align and Translate*. ICLR.
- Luong, M.-T., Pham, H., & Manning, C. D. (2015). *Effective Approaches to Attention-based Neural Machine Translation*. EMNLP.
- Yang, M., Li, X., & Liu, Y. (2021). *Sequence-to-Point Learning Based on an Attention Neural Network for Nonintrusive Load Decomposition*. Electronics, 10(14), 1657.