Improving IMU-Based Online Handwriting Recognition: Upgrading from CTC to Attention-based Autoregressive Decoder

Muhammad Ajlal Khan, Matriculation: 23456620

***Abstract*—Inertial measurement unit (IMU) time-series offer a practical sensing modality for online handwriting recognition (HWR), but reliable sequence modeling must handle variable-length inputs, writer variability, and error modes that are not well characterized by aggregate metrics alone. This project studies a transition from a strong CTC-based baseline to an attention-based autoregressive (AR) encoder–decoder for IMU-HWR, and evaluates architectural and training choices on internal word- and sentence-level datasets. To enable efficient Transformer decoding on highly variable sequence lengths, we implement batch-wise rectangularization with masking, avoiding global fixed-length padding while preserving GPU efficiency. Empirically, AR decoding substantially improves word-level recognition as reflected by WER, while exhibiting a different CER–WER trade-off than CTC. We further introduce lightweight gating at the scaled dot-product attention (SDPA) output and observe consistent improvements across tasks, indicating that modulating attention outputs can stabilize generation and reduce errors without significant computational overhead. In contrast, BPE-based tokenization does not yield consistent gains over character-level decoding in this setting. Beyond headline scores, distributional analysis reveals strongly bimodal behavior (a large mass of exact matches) and a heavy-tailed nonzero error regime; targeted collision and writer-stratified analyses suggest that frequent-label bias and writer shift contribute to a subset of catastrophic failures. Qualitative inspection using cross-attention heatmaps and Grad-CAM1D supports this picture by highlighting alignment fragmentation and attention collapse in high-severity errors. Overall, the results establish a practical AR Transformer pipeline for IMU-HWR and motivate thesis directions centered on tail-risk reduction, collision-aware analysis, decoding improvements, and robustness to writer-specific shift.**

***Index Terms*—IMU-based, Transformer, Tokenization, CTC**

I. Introduction

Online handwriting recognition (HWR) from inertial measurement unit (IMU) time series enables free-form text input using sensor-equipped pens on arbitrary surfaces, without relying on cameras or dedicated digitizers. While IMU-HWR is lightweight and privacy-friendly, writer-independent (WI) recognition remains challenging due to inter-writer variability (stroke dynamics, pen handling, sensor placement) and sensor noise, which together induce substantial distribution shift across users.

A strong WI baseline in this setting is the REWI architecture, which combines a compact CNN encoder with a BiLSTM decoder trained using connectionist temporal classification (CTC). CTC avoids explicit alignments but introduces a specialized decoding pipeline (blank modeling and collapse rules) and may limit how strongly predictions leverage long-range conditional structure. Motivated by this, the goal of this project is to migrate from the CTC pipeline to an attention-based autoregressive (AR) sequence-to-sequence formulation using a Transformer decoder trained with teacher forcing and cross-entropy. A key practical obstacle is that IMU traces yield variable-length encoder sequences, whereas Transformer implementations typically assume rectangular tensors; we address this via dynamic batch-wise rectangularization, padding encoder outputs only to the batch maximum and masking padded positions during attention.

Building on this mechanism, we conduct a controlled experimental progression on internal STABILO word- and sentence-level datasets: (i) replace the BiLSTM with a Transformer while retaining CTC training to isolate architectural effects, (ii) train the Transformer autoregressively with character targets, (iii) evaluate BPE tokenization variants, and (iv) integrate lightweight post-SDPA gating (headwise and elementwise). Beyond aggregate CER/WER, we characterize error structure using normalized per-sample edit distance distributions, collision and writer-stratified analyses, and qualitative inspection via cross-attention heatmaps and Grad-CAM1D. The remainder of the report is organized as follows: Section II reviews related work; Section III describes the proposed methods and metrics; Section IV details the datasets; Section V presents results and diagnostic analyses; Section VI discusses implications and thesis directions; and Section VII concludes the report.

II. Literature Review

A. CTC-based sequence labeling

CTC is a standard objective for unsegmented sequence labeling, widely used in speech and handwriting recognition. It avoids explicit alignments by marginalizing over monotonic alignments between input frames and output labels, typically decoded with greedy or beam-search post-processing. [3]

CTC simplifies training when alignments are unknown, but introduces an auxiliary decoding pipeline (blank modeling, collapsing repeats) and may constrain the modeling of strong conditional dependencies across output tokens.

B. Attention-based sequence-to-sequence and Transformers

Transformer architectures replace recurrence with self-attention, improving parallelism and enabling global context modeling. [10] For seq2seq, Transformers are commonly trained autoregressively with teacher forcing, using a causal mask to prevent attending to future target tokens.

C. Tokenization with BPE

Subword tokenization (e.g., byte-pair encoding, BPE) provides an open-vocabulary representation by encoding words as sequences of learned subword units. [8] In HWR, subword tokenization may reduce decoding length and potentially improve word-level accuracy, but can also introduce new failure modes (token boundary errors, mismatch between acoustic/kinematic evidence and subword segmentation).

D. Gated attention

Recent work shows that inserting simple sigmoid gating into attention can improve stability and performance. In particular, applying head-specific gating after scaled dot-product attention (SDPA) output ("G1") consistently improves large-model training behavior; both headwise and elementwise variants have been studied. [6]

III. Methods

A. Replacing BiLSTM with Transformer under CTC training

As an intermediate step, we replace the BiLSTM decoder with a comparable-sized Transformer decoder module while retaining the original CTC objective (character-based recognition). This isolates architectural effects from the objective change. This is our CTC baseline and we will refer to it in the next parts of the report as Ours-B-CTC.

B. Autoregressive training via batch-wise rectangularization

To enable autoregressive Transformer decoding with variable-length encoder outputs, we apply **dynamic batch-wise rectangularization**:

1) For a minibatch, compute encoder output lengths $\{T_i\}_{i=1}^{B}$.
2) Let $T_{\max} = \max_i T_i$ within the batch.
3) Right-pad each encoder sequence to length $T_{\max}$ (not a fixed global length).
4) Construct a key-padding mask so the decoder attention ignores padded positions.

This preserves efficient batched execution while avoiding the heavy-handed global padding strategy (e.g., padding all inputs to 1024) that is sometimes used when models require fixed-length inputs. [5]

Decoder training uses standard teacher forcing: the target sequence is shifted right with a start token, and a causal mask prevents attention to future target positions. This is our AR baseline and we will refer to it in the next parts of the report as Ours-B-AR.

C. Character vs. BPE decoding

We evaluate whether BPE tokenization improves performance over character-level decoding by training three BPE vocabularies (100/200/300 merge operations; denoted BPE100/200/300) for both word and sentence tasks, keeping architecture and training procedure fixed.

D. Gated attention (SDPA output gating)

Inspired by [6], we integrate two gating variants after the SDPA output in each attention head:

- **Elementwise gating (G1, SDPA elementwise):** per-head, per-feature sigmoid gate $g_h = \sigma(W_h y_h + b_h)$ applied as $y_h \leftarrow y_h \odot g_h$.
- **Headwise gating (G1, SDPA headwise):** per-head sigmoid gate producing a scalar per token, $g_h = \sigma(w_h^\top y_h + b_h)$, applied as $y_h \leftarrow y_h \cdot g_h$.

All gating experiments are trained autoregressively with character-based targets.

E. Metrics

a) Notation.: For sample i, let y_i be the ground truth and $\hat{y}_i$ the prediction. Let $d(\cdot,\cdot)$ denote Levenshtein edit distance.

b) Transcription quality (headline).: **CER** is computed at the character level and **WER** at the word level (after tokenizing on whitespace). We report standard corpus-level (micro) rates:

$$\mathrm{CER} = \frac{\sum_i d_{\mathrm{char}}(y_i, \hat{y}_i)}{\sum_i |y_i|}, \qquad \mathrm{WER} = \frac{\sum_i d_{\mathrm{word}}(y_i, \hat{y}_i)}{\sum_i |y_i|_{\mathrm{word}}}.$$

c) Per-sample normalized error (distributional analysis).: We additionally use the per-sample normalized character error

$$e_i = \frac{d_{\text{char}}(y_i, \hat{y}_i)}{|y_i|},$$

from which we report: exact-match rate $P(e = 0)$, tail probabilities $P(e > \tau)$, and quantiles (including conditional quantiles computed over $e > 0$). We summarize e via a micro-average $\sum_i d_i / \sum_i |y_i|$ and a macro-average $\frac{1}{N} \sum_i e_i$.

d) Length dependence.: We measure Pearson correlations between target length and both raw distance and normalized error: $\rho(d, |y|)$ and $\rho(e, |y|)$, reported for all samples and for errors only ($e > 0$).

e) Collision diagnostics.: A *collision* occurs when an incorrect prediction exactly matches the ground-truth label of a different sample. For collided predictions $\hat{y}$, we compute $gt\_label\_count(\hat{y})$ (frequency of $\hat{y}$ in the ground-truth corpus). For writer-stratified analysis we use N_w (samples), E_w (errors), C_w (colliding error samples), and derived rates: E_w/N_w, C_w/N_w, and C_w/E_w. Collision concentration is summarized by $\text{lift}(K)$ = collision share / sample share for top-K writers.

f) Input-similarity baseline for collisions.: For curated collision pairs (x_i, x_j) we compute cosine similarity between dynamics-feature vectors: $sim_{\text{collision}} = \cos(f(x_i), f(x_j))$. We compare against a within-task, within-fold random baseline $sim_{\text{random}} = \cos(f(x_i), f(x_k))$ with $k \neq i$, reporting $\Delta = sim_{\text{collision}} - sim_{\text{random}}$ and $P(\Delta > 0)$.

g) Efficiency.: We report #Params and MACs as recorded in experiment logs.

F. Concatenation-based data augmentation

To encourage robustness to longer temporal contexts and expose the autoregressive decoder to extended input streams, we evaluated a concatenation-based augmentation strategy on Ours-B-AR + ElementwiseGating model. We construct "virtual" training samples by concatenating K IMU sequences along the time axis and concatenating their label strings, *without* inserting an explicit separator token; minibatches are padded to the maximum input and target lengths within each batch. We tested two variants: **random-writer** concatenation (mixing writers within a stream) and **same-writer** concatenation (all subsequences from the same writer via id_writer. For both variants, we used $\texttt{max\_T} = 4096$ time steps for the word task and $\texttt{max\_T} = 16384$ for the sentence task. Table IX summarizes the results.

IV. Datasets

All experiments are conducted on writer-independent (WI) split of the private/non-disclosed STABILO word and sentence datasets. Data are recorded with an IMU-equipped pen producing 13 channels (two accelerometers, gyroscope, magnetometer, force sensor) at 100 Hz. [5] The word dataset comprises 54,666 English/German word samples collected in 984 recording sessions, covering 59 character categories (upper/lowercase across both languages). [5]

In addition, an internal sentence-based dataset is used for multi-word recognition, following the evaluation setting described in REWI for sentence-level recognition. [5]

V. Results

In the **Experimental Results** section, we summarize the outcomes of all model variants and training configurations, emphasizing the progression from the CNN–BiLSTM–CTC based baseline to our CNN–AR–Transformer models and the main ablations (autoregressive training, BPE tokenization, and SDPA-output gating). Unless stated otherwise, all experiments use character-based targets. In the **Quantitative Analysis** section, we report per-sample normalized Levenshtein error $e = d/|y|$ together with micro/macro averages and quantiles. The resulting distributions are strongly *bimodal*: a large mass of exact matches ($e = 0$) coexists with a heavy-tailed nonzero error regime that captures the severity spectrum among incorrect predictions. We then investigate a concrete failure pattern via **collision analysis**, where an incorrect prediction exactly matches the ground-truth label of another sample. Collisions are common in both datasets and are disproportionately associated with high-frequency labels; moreover, curated collision pairs show little systematic increase in IMU similarity relative to random baselines, while writer-stratified results reveal that collisions are concentrated in a subset of writers. Finally, in the **Qualitative Analysis** section, we inspect representative *correct*, *typical error* ($p50$ over $e > 0$), *moderate* ($p90$), and *severe* ($p99$) cases using decoder cross-attention heatmaps and 1D Grad-CAM over the encoder time axis. These visualizations suggest that typical errors often retain partial alignment but exhibit repeated attention anchors, whereas severe failures more frequently involve fragmented or collapsed attention patterns, consistent with a breakdown in token–time alignment.

A. Experimental Results

1) Reference baselines: Table I reports the REWI baseline results (CNN–BiLSTM–CTC) on internal STABILO word and sentence data.

2) Transformer decoder with CTC objective: Table II shows the intermediate experiment where the BiLSTM is replaced by a Transformer decoder while keeping CTC training and character-based recognition.

TABLE I: REWI baseline results (CTC).

Model	CER	WER	#Params (M)	MACs
Rewi-B (Word)	9.39	31.81	4.64	695 M
Rewi-B (Sent)	6.54	23.51	4.64	2.78 B

TABLE II: Replacing BiLSTM with Transformer under the original CTC pipeline.

Model	CER	WER	#Params (M)	MACs
Ours-B-CTC (Word)	8.70	31.21	4.69	430 M
Ours-B-CTC (Sent)	7.89	24.71	4.69	1.7 B

3) Autoregressive Transformer decoding (character-based): After enabling batch-wise rectangularization (Section III-B), we train the Transformer autoregressively with character-level targets. Results are in Table III.

TABLE III: Autoregressive Transformer decoding with character-based targets.

Model	CER	WER	#Params (M)	MACs
Ours-B-AR (Word)	12.80	20.60	4.69	430 M
Ours-B-AR (Sent)	11.35	15.83	4.69	1.7 B

4) BPE tokenization experiments: Table IV and Table V summarize BPE experiments (BPE100/200/300). No consistent improvement over decoding is observed.

TABLE IV: Autoregressive decoding on the word dataset with BPE tokenization.

Model	CER	WER	#Params (M)	MACs
Ours-B-AR + BPE100	13.01	21.09	4.69	430 M
Ours-B-AR + BPE200	13.06	20.90	4.69	430 M
Ours-B-AR + BPE300	13.02	20.90	4.69	430 M

5) Gated attention experiments: Table VI and Table VII report the two SDPA-output gating variants. Both provide clear gains over the ungated AR Transformer, especially for .

6) Consolidated comparison: Table VIII condenses the full story into a single view (word vs. sentence), to fascilitate a quick overview.

7) Concatenation-based data augmentation: Table IX shows that performance degrades markedly on both datasets, indicating that naive long-context concatenation is not beneficial under the current architecture and decoding setup.

B. *Quantitative Analysis*

a) Metric.: We evaluate transcription quality using the *normalized Levenshtein distance*

$$e_i = \frac{d_i}{|y_i|}, \tag{1}$$

TABLE V: Autoregressive decoding on the sentence dataset with BPE tokenization.

Model	CER	WER	#Params (M)	MACs
Ours-B-AR + BPE100	11.11	15.50	4.69	1.7 B
Ours-B-AR + BPE200	11.10	15.44	4.69	1.7 B
Ours-B-AR + BPE300	11.13	15.60	4.69	1.7 B

TABLE VI: Autoregressive decoding on the word dataset with gated attention (character-based).

Model	CER	WER	#Params (M)	MACs
Ours-B-AR + ElementwiseGating	10.37	17.43	4.69	430 M
Ours-B-AR + HeadwiseGating	10.29	17.54	4.69	430 M

where d_i is the character-level Levenshtein distance between prediction and ground-truth, and $|y_i|$ is the ground-truth character length. This normalization yields a length-aware error rate per sample and provides a continuous distribution that is comparable across word- and sentence-level tasks. We report both a *micro-average* $\sum_i d_i / \sum_i |y_i|$ and a *macro-average* $\frac{1}{N}\sum_i e_i$, together with quantiles of the per-sample distribution $\{e_i\}$. For visualization (histograms/CDFs), we clip the x-axis at $e \leq 1.5$ for readability because a small number of outliers extend beyond this range.

b) Aggregate behavior.: Table X shows that both datasets exhibit a pronounced mass at $e = 0$ (exact matches), while the nonzero errors form a heavy-tailed distribution. The micro-averaged normalized error is 0.13 (Word) and 0.16 (Sentence), with macro averages of 0.13 and 0.17, respectively. The tail probability $P(e > 1)$ is 1.12 (Word) and 4.38 (Sent), which is low for Word-model but non-negligible for Sent-model, indicating a small set of severe failures. Interestingly, the tail probability $P(e > 0.5)$ is much higher at around 12.03 (Word) and 16.93 (Sent), indicating a large distribution of errors falling in moderate error range.

c) Fold-wise variability.: Table XI indicates moderate variability across folds, but no fold collapse under the normalized metric. Tail behavior is stable across folds, with $P(e > 1)$ remaining in the $[4.20\%, 5.65\%]$ range for Senetnce task and in the $[0.80\%, 1.45\%]$ range for Word task. Similalry, with $P(e > 0.5)$ remaining in the $[14.80\%, 18.99\%]$ range for Senetnce task and in the $[11.15\%, 12.65\%]$ range for Word task.

d) Representative examples near quantiles.: To describe the tail behavior without imposing ad-hoc buckets, we compute empirical quantiles of the *nonzero* normalized errors (i.e., conditioning on $e > 0$). For example, the nonzero p_{90} is a threshold such that 90% of *incorrect* predictions have $e \leq p_{90}$, and 10% lie above it. To ground the quantiles in concrete behavior, Table XII

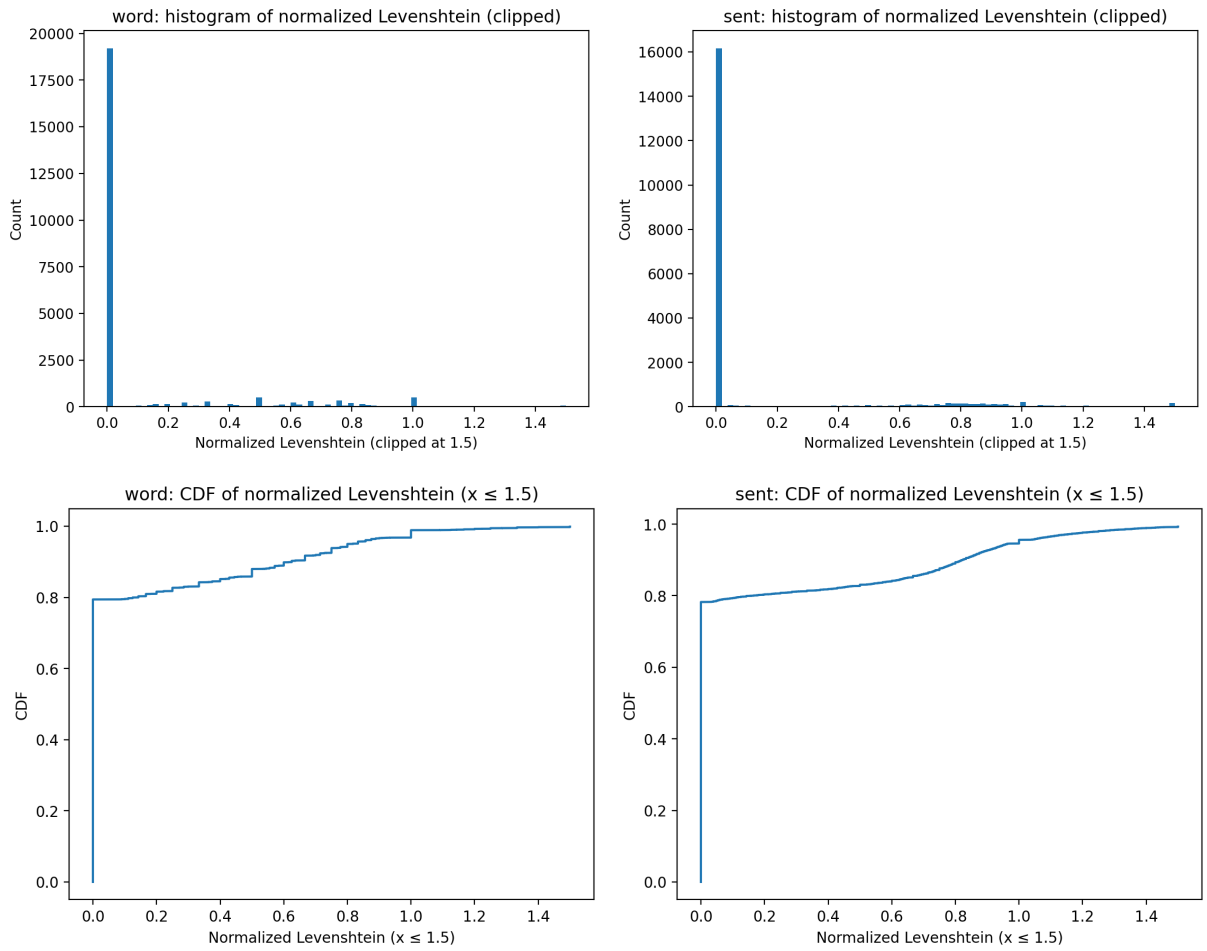


Fig. 1: Distribution of normalized Levenshtein error $e = d/|y|$ with x-axis clipped to $e \leq 1.5$ for readability. Top: histograms. Bottom: CDFs.

TABLE VII: Autoregressive decoding on the sentence dataset with gated attention (character-based).

Model	CER	WER	#Params (M)	MACs
Ours-B-AR + ElementwiseGating	8.68	12.30	4.69	1.7 B
Ours-B-AR + HeadwiseGating	8.68	12.25	4.69	1.7 B

shows representative samples whose normalized error $e = d/|y|$ is closest to p_{50}, p_{90}, and p_{99}, with two examples per quantile and dataset. These quantiles are computed on the *nonzero* subset ($e > 0$, equivalently $d > 0$), so that the selected samples reflect *error severity* among incorrect predictions rather than being dominated by exact matches, as in our case, $p_{50} = 0$ for both Word and Sentence tasks.

e) Length dependence (raw vs. normalized; all vs. errors-only).: Table XIII reports Pearson correlations between ground-truth length $|y|$ and both raw Levenshtein distance d and normalized error $e = d/|y|$, computed on (i) the full validation set and (ii) incorrect predictions only ($d > 0$). Over all samples, d shows a small positive correlation with $|y|$ (longer targets admit more absolute edits), while e is near zero, indicating that relative error is largely length-invariant. When conditioning on errors ($d > 0$), the correlation between d and $|y|$ becomes strongly positive, whereas the correlation between e and $|y|$ becomes slightly negative, suggesting that among failures, longer labels tend to have proportionally milder errors. Overall, normalization substantially reduces length dependence and supports using e for cross-sample and cross-task comparability.

C. *Collision analysis*

1) Frequency bias vs. input similarity: To diagnose whether *raw collisions* (i.e., cases where a misprediction exactly matches the ground-truth string of a different sample) arise from input-space confusion or from label-

TABLE VIII: Consolidated results across experiments on STABILO internal datasets.

Task	Model	CER	WER	#Params (M)	MACs
Word	Rewi-B-CTC	9.39	31.81	4.64	695 M
	Ours-B-CTC	8.70	31.21	4.69	430 M
	Ours-B-AR	12.80	20.60	4.69	430 M
	Ours-B-AR + BPE100	13.01	21.09	4.69	430 M
	Ours-B-AR + BPE200	13.06	20.90	4.69	430 M
	Ours-B-AR + BPE300	13.02	20.90	4.69	430 M
	Ours-B-AR + Elementwise Gating	10.37	17.43	4.69	430 M
	Ours-B-AR + Headwise Gating	10.29	17.54	4.69	430 M
Sentence	Rewi-B-CTC	6.54	23.51	4.64	2.78 B
	Ours-B-CTC	7.89	24.71	4.69	1.7 B
	Ours-B-AR	11.35	15.83	4.69	1.7 B
	Ours-B-AR + BPE100	11.11	15.50	4.69	1.7 B
	Ours-B-AR + BPE200	11.10	15.44	4.69	1.7 B
	Ours-B-AR + BPE300	11.13	15.60	4.69	1.7 B
	Ours-B-AR + Elementwise Gating	8.68	12.30	4.69	1.7 B
	Ours-B-AR + Headwise Gating	8.68	12.25	4.69	1.7 B

TABLE IX: Concatenation-based augmentation on Ours-B-AR model.

Task	Variant	CER	WER
Word	Random-writer	17.42	33.40
Word	Same-writer	18.74	38.67
Sentence	Random-writer	–	–
Sentence	Same-writer	22.93	32.36

frequency/decoder effects, we perform two complementary checks.

a) Check 1: collision label-frequency concentration.: We quantify how often collided predictions correspond to frequently occurring ground-truth labels. For each collision event, we compute $\texttt{gt\_label\_count}(\hat{y})$, the number of times the collided prediction string $\hat{y}$ appears as a ground-truth label in the corresponding dataset (Word vs. Sentence). Table XIV summarizes the resulting distribution using both (i) a *UID-weighted* view (one entry per colliding error sample, avoiding inflation by repeated matches) and (ii) an *event-weighted* view (each collision match counted). In both datasets, collided predictions are typically high-frequency labels (median $\texttt{gt\_label\_count} \approx 14$–$18$; upper tails reach ≈ 23 for Sentence and ≈ 28 for Word), indicating that collisions disproportionately involve labels that are repeated many times in the corpus.

b) Check 2: input similarity on curated collision pairs.: To test whether collisions reflect systematic input-space confusion, we curate collision pairs (x_i, x_j) and compare their IMU similarity to a within-task, within-fold random baseline (x_i, x_k), where k is sampled uniformly from the same validation fold with $k \neq i$ (fixed seed). Similarity is computed as cosine similarity between dynamics-based feature vectors (first-order differences and inter-channel correlations), which remain informative under per-sample normalization. Across 200 curated pairs per dataset, the mean similarity gap $\Delta = \mathrm{sim}(x_i, x_j) - \mathrm{sim}(x_i, x_k)$ is small (Sentence: $\Delta_{\text{mean}} \approx 0.012$; Word: $\Delta_{\text{mean}} \approx -0.002$; see `baseline_similarity_summary_new.csv`). Consistently, the fraction of cases with $\Delta > 0$ is close to chance (Sentence: 0.515; Word: 0.475). Overall, these results provide little evidence that collisions are primarily driven by consistent input-space similarity; instead, collisions appear compatible with weak or inconsistent conditioning and label-frequency/decoder-prior effects.

2) Writer-based collision analysis: While the previous collision analyses indicated a strong label-frequency component (collided predictions tend to be frequent ground-truth strings) and little evidence for systematic input-space similarity beyond a random baseline, collisions may still be *heterogeneous across writers*. In IMU handwriting, writer-specific motion characteristics (sensor placement, stroke speed, and style) can induce distribution shifts that affect how model errors manifest. We therefore analyze collisions stratified by writer to determine whether a small subset of writers accounts for a disproportionate share of collisions.

a) Writer mapping.: Each validation sample is identified by $(\texttt{task}, \texttt{fold}, \texttt{sample\_index})$ in the unified prediction CSV. The dataset annotation files (`val.json`) provide a writer identifier `id_writer` for each entry within each fold. Because `sample_index` corresponds to the index into `val.json`'s annotation list for the given fold, we can deterministically map every sample to its writer:

$$(\texttt{task}, \texttt{fold}, \texttt{sample\_index}) \mapsto \texttt{id\_writer}.$$

We attach `id_writer` both to (i) the full validation set (to measure per-writer sample counts and error rates) and (ii) the collision set (to measure per-writer collision rates).

TABLE X: Overall quantitative summary using normalized Levenshtein distance $e_i = d_i/|y_i|$. Micro denotes $\sum_i d_i / \sum_i |y_i|$; Macro denotes $\frac{1}{N}\sum_i e_i$. Quantiles are computed over per-sample e_i. Here $\overline{|y|}$ is the mean number of characters per ground-truth label and $\overline{|y|}_{\mathrm{w}}$ is the mean number of words per ground-truth label.

Dataset	N	Exact (%)	$\overline{\lvert y\rvert}$	$\overline{\lvert y\rvert}_{\mathrm{w}}$	Micro	Macro	p_{50}	p_{90}	p_{95}	p_{99}	$P(e>0.5)$ (%)	$P(e>1)$ (%)
Word	24163	79.41	6.53	1	0.13	0.13	0	0.63	0.8	1.14	12.03	1.12
Sent	20639	78.24	18.93	2.97	0.16	0.17	0	0.82	1.0	1.43	16.93	4.38

TABLE XI: Fold-wise summary with normalized Levenshtein distance. Micro denotes $\sum_i d_i / \sum_i |y_i|$ within each fold; p_{50}/p_{90}/p_{99} are computed over per-sample normalized errors within the fold. The results are based on the complete validation dataset (including $e = 0$).

Dataset	Fold	N	Exact (%)	Micro	p_{90}	p_{99}	$P(e>0.5)$ (%)	$P(e>1)$ (%)
Word	0	4734	81.26	0.12	0.60	1.14	11.15	1.14
Word	1	5002	79.29	0.13	0.67	1.00	12.40	0.80
Word	2	4694	77.42	0.14	0.67	1.25	12.65	1.45
Word	3	4656	79.77	0.13	0.63	1.20	12.29	1.33
Word	4	5077	79.30	0.13	0.60	1.00	11.68	0.91
Sentence	0	4188	79.08	0.16	0.81	1.39	16.07	4.20
Sentence	1	3684	77.99	0.17	0.83	1.34	17.21	3.99
Sentence	2	4416	77.94	0.17	0.81	1.39	17.28	4.26
Sentence	3	3871	81.30	0.14	0.78	1.35	14.80	3.64
Sentence	4	4480	75.29	0.19	0.87	1.60	18.99	5.65

TABLE XII: Representative examples near quantiles of the *nonzero* normalized Levenshtein error distribution (e) (two samples per quantile and dataset). Quantiles are computed on $e > 0$ (i.e., incorrect predictions only).

Dataset	Quantile	Fold	Idx	d	e	Prediction	Ground truth
Word	p_{50}	0	236	3	0.60	Steuer	Staus
Word	p_{50}	0	1001	3	0.60	vorne	renne
Word	p_{90}	0	145	8	1.00	Durste	ganze
Word	p_{90}	0	175	5	1.00	Dinosaurier	Pflaster
Word	p_{99}	0	2809	12	1.50	Meerschweinchen	Reaktion
Word	p_{99}	0	1406	6	1.50	Ödland	quer
Sentence	p_{50}	0	1423	12	0.80	not ony other boxy	round of boxing
Sentence	p_{50}	0	1946	12	0.80	to inght way	mountain biking
Sentence	p_{90}	0	130	16	1.23	Was um alles in der We	Wait and see!
Sentence	p_{90}	0	1443	16	1.23	SMS; Kurznachrich	single ticket
Sentence	p_{99}	4	2179	16	1.78	Freundschaft schließ	free time
Sentence	p_{99}	0	3006	25	1.79	Klatsch/Klatscht in die Hände.	klar; deutlich

TABLE XIII: Pearson correlation between ground-truth length $|y|$ and error: raw Levenshtein d vs. normalized $e = d/|y|$. Results are shown for the full set (All) and for incorrect predictions only ($d > 0$) to separate overall length association from conditional error severity.

Dataset	$\rho(d, \lvert y\rvert)$ (All)	$\rho(e, \lvert y\rvert)$ (All)	$\rho(d, \lvert y\rvert)$ $(e > 0)$	$\rho(e, \lvert y\rvert)$ $(e > 0)$
Sentence	0.140	-0.032	0.664	-0.080
Word	0.203	0.006	0.624	-0.105

TABLE XIV: Distribution of `gt_label_count`$(\hat{y})$ for collided predictions $\hat{y}$ (how often the collided string appears as ground-truth in the dataset). UID-weighted: one row per colliding error sample. Event-weighted: one row per collision event.

Dataset	View	n	Median	p_{90}	p_{95}
Sentence	UID-weighted	1712	15	21	23
Word	UID-weighted	2845	14	26	29
Sentence	Event-weighted	24467	17	23	25
Word	Event-weighted	41576	19	31	40

b) Definitions.: For each writer w, we compute:

- N_w: number of validation samples (`n_samples`),
- E_w: number of erroneous predictions (`n_errors`, defined by raw Levenshtein distance $d > 0$ between prediction and ground truth),
- C_w: number of *colliding error samples* (`n_collision_uids`), i.e., unique error samples whose prediction matches the ground-truth string of at least one other sample.

From these counts we derive:

$$\texttt{error\_rate} = \frac{E_w}{N_w},$$
$$\texttt{collision\_rate} = \frac{C_w}{N_w},$$
$$\texttt{collision\_given\_error} = \frac{C_w}{E_w}.$$

To avoid inflating writer effects due to labels repeated many times, we treat collisions primarily in a *UID-weighted* manner (one entry per colliding error sample), and additionally summarize *event-weighted* collision counts where appropriate.

c) Concentration analysis (top-K writers).: To quantify whether collisions are concentrated in a small subset of writers, we rank writers by C_w and compute the share of all colliding error samples attributable to the top-K writers. To control for dataset imbalance across writers, we also compute the share of all samples contributed by the same top-K writers and report the *lift*:

$$\texttt{lift}(K) = \frac{\text{collision\_share}(K)}{\text{sample\_share}(K)}.$$

Table XV shows that collisions are substantially over-represented among the top writers in both datasets. For example, for Sentence the top-10 writers account for 14.43% of colliding samples while representing only 4.82% of the data (lift $\approx$ 2.99). For Word, the top-10 writers account for 15.22% of colliding samples vs. 7.58% of the data (lift $\approx$ 2.01). This indicates a meaningful writer-specific component: certain writers are *collision-prone* beyond what would be expected from their sample counts alone.

d) Per-writer variability.: In addition to concentration, we observe substantial variability across writers in `collision_given_error`, i.e., the fraction of a writer's errors that become collisions. Table XV shows that among top-10 error-prone writers, this value reaches $\approx$ 54% (Sent) and $\approx$ 52% (Word) suggesting that collisions are not purely a global phenomenon; rather, writer-specific style or distribution shift modulates the collision tendency. Importantly, this writer effect complements (rather than contradicts) the earlier frequency-bias finding: collisions are globally associated with frequent labels, but a subset of writers contributes disproportionately to these collision outcomes.

D. *Qualitative analysis: cross-attention and temporal saliency (word-level)*

To complement the quantitative evaluation on the full validation set (Section V-B), we qualitatively inspect *where* the AR decoder focuses in the input time-series when generating output tokens. We analyze (i) decoder-to-encoder cross-attention heatmaps and (ii) encoder-side temporal saliency via Grad-CAM1D. The aim is not to explain the model exhaustively, but to provide interpretable evidence for typical behaviors in correct predictions and in failures of increasing severity under the normalized edit distance $e = d/|y|$.

a) Sample selection (severity regimes).: We select a small set of representative validation samples from four regimes: (i) **Correct**: $e = 0$, (ii) **Typical error**: $e > 0$ near the conditional median ($p50$ over the nonzero errors), (iii) **Moderate error**: $e > 0$ near the upper tail ($p90$ over the nonzero errors), (iv) **Severe error**: extreme failures in the heavy tail (e.g., $p99$ over the nonzero errors or $e > 1$). This regime definition matches the distribution in Table XI: most samples are exact matches ($e = 0$), while incorrect predictions span a heavy-tailed spectrum of severities.

b) Cross-attention visualization.: For each selected sample, we extract decoder-to-encoder cross-attention during greedy decoding, average across heads and layers, and visualize the resulting matrix in $\mathbb{R}^{T_{\text{tok}} \times T_{\text{enc}}}$. Because the encoder is temporally downsampled, we do not expect a perfectly diagonal band; rather, *consistent* token-dependent focus and a coherent progression across encoder time indicate stable alignment, whereas strong jumps, repeated vertical stripes (attention repeatedly returning to the same encoder positions), or widespread instability suggest weaker alignment.

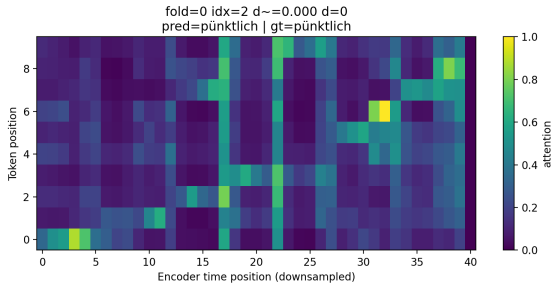
c) Grad-CAM for 1D time-series.: We complement attention with Grad-CAM1D computed on a late encoder convolutional layer by backpropagating the score of the predicted sequence and aggregating gradients into a temporal importance curve. The curve is upsampled to the input length and shown alongside the input RMS. Since gradient-based saliency can saturate (and its amplitude depends on the chosen scaling), we interpret Grad-CAM primarily as a *relative* indicator of which time segments support the prediction, not as a calibrated measure of causal contribution.

1) Illustrative patterns across regimes: Across the inspected samples, increasing error severity is associated with less reliable token–time alignment: attention either becomes fragmented (frequent jumps) or collapses onto a small set of encoder positions. Grad-CAM can remain localized (a few peaks) or become broadly distributed depending on the sample, but severe errors often coincide with attention patterns that fail to track distinct temporal evidence for successive tokens.

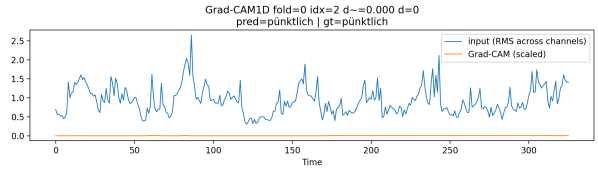
a) Correct predictions ($e = 0$).: In the correct example (Fig 2), cross-attention shows recurring, structured focus on a small number of encoder regions, indicating that the decoder consistently reuses specific temporal evidence while emitting characters. In this case, the Grad-CAM curve is nearly flat after scaling, which is compatible with gradient saturation in high-confidence predictions; therefore, attention provides the

TABLE XV: Collision concentration among the top-K writers. "Collision share" is the fraction of all colliding error samples contributed by the top-K writers; "sample share" is the fraction of all samples contributed by the same writers. Lift > 1 indicates over-representation of collisions among those writers. CGR: collision_given_error.

Dataset	K	Collision share (%)	Sample share (%)	Lift	CGR
Sentence	5	8.64	2.50	3.46	0.495
Sentence	10	14.43	4.82	2.99	0.543
Sentence	20	23.95	7.72	3.10	0.544
Word	5	9.17	4.12	2.23	0.517
Word	10	15.22	7.58	2.01	0.522
Word	20	25.13	10.67	2.35	0.581



(a) Cross-attention (averaged across heads/layers).



(b) Grad-CAM1D overlaid on input RMS.

Fig. 2: **Correct prediction** (fold 0, idx 2): `pünktlich`→`pünktlich` ($e = 0$).

more informative alignment signal for this regime.

b) Typical errors (near $p50$ over $e > 0$).: For this case (idx 1001, `vorne` vs. `renne`), the cross-attention heatmap (Fig. 3(a)) shows pronounced emphasis on an early segment and repeated vertical bands at a few encoder time indices across multiple tokens. This suggests partial or unstable alignment, where the decoder repeatedly falls back to a small set of anchor positions instead of progressing smoothly through time. The corresponding Grad-CAM1D trace (Fig. 3(b)) highlights early activity but then becomes comparatively uniform across a long interval, indicating that the encoder evidence is not sharply localized into a few decisive segments for this prediction.

c) Moderate errors (near $p90$ over $e > 0$).: For the $p90$ example (idx 145, `Dinosaurier` vs. `Pflaster`), attention becomes more fragmented (Fig. 3(c)): multiple token-specific hotspots appear, but with noticeable jumps and partial reuse of the same encoder regions rather than a coherent progression. Grad-CAM shows several distinct peaks (Fig. 3(d)), suggesting that multiple temporally separated segments contribute; however, the decoder fails to translate this evidence into a stable token–time alignment, leading to a substantially wrong word.

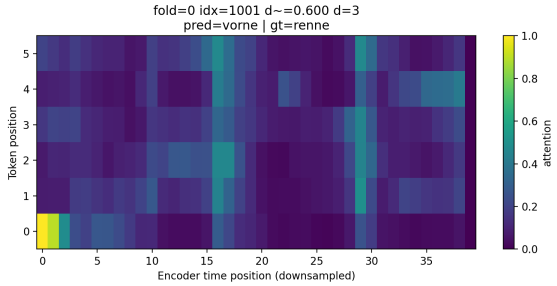
d) Severe errors (near $p99$ over $e > 0$).: In the most severe example (idx 1406, `Ödland` vs. `quer`), cross-attention collapses onto a dominant encoder position (Fig. 3(e)), visible as a bright vertical stripe shared across many tokens, with only secondary attention elsewhere. This collapse indicates that the decoder effectively ignores most temporal evidence and repeatedly conditions on a narrow encoder fragment. Grad-CAM (Fig. 3(f)) still marks several mid-sequence regions as salient, but the mismatch between broad encoder-side saliency and collapsed decoder attention points to a breakdown in token–time alignment rather than a simple lack of informative input.

VI. Discussion

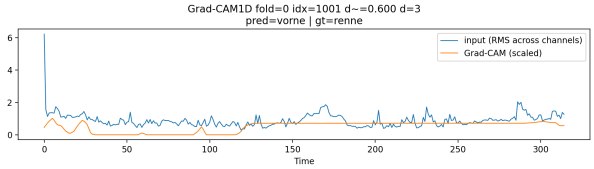
This project investigated a practical migration path from a CTC-based online handwriting recognition (HWR) system for IMU time series toward an attention-based autoregressive (AR) sequence-to-sequence formulation. Across internal STABILO word- and sentence-level datasets (13 IMU channels at 100 Hz, writer-independent split), the central empirical finding is that AR decoding—enabled by dynamic batch-wise rectangularization—substantially improves word-level transcription quality as reflected by WER, and that simple post-SDPA output gating further improves both tasks with consistent gains.

A. What changed when moving from CTC to AR, and why it matters

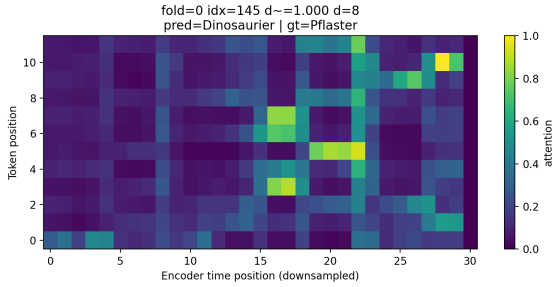
The results highlight a clear metric trade-off between CTC and AR training. On the word task, the CTC baseline achieves relatively low CER but very high WER, whereas AR decoding increases CER but dramatically reduces WER (see Table VIII) (e.g., Rewi-B-CTC: CER 9.39, WER 31.81 vs. Ours-B-AR: CER 12.80, WER



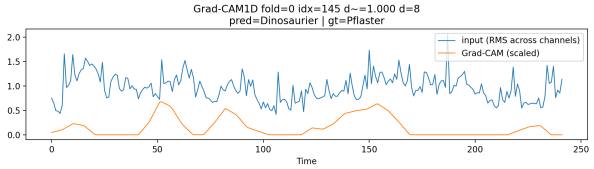
(a) $p50$ **over** $e > 0$ (fold 0, idx 1001): `vorne` vs. `renne` ($e = 0.6$). Cross-attention.



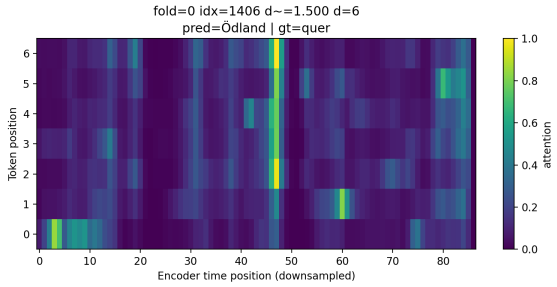
(b) $p50$ **over** $e > 0$ (fold 0, idx 1001). Grad-CAM1D.



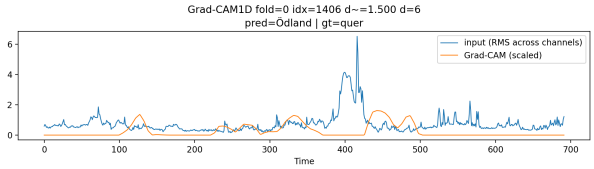
(c) $p90$ **over** $e > 0$ (fold 0, idx 145): `Dinosaurier` vs. `Pflaster` ($e = 1.0$). Cross-attention.



(d) $p90$ **over** $e > 0$ (fold 0, idx 145). Grad-CAM1D.



(e) $p99$ **over** $e > 0$ (fold 0, idx 1406): `Ödland` vs. `quer` ($e = 1.5$). Cross-attention.



(f) $p99$ **over** $e > 0$ (fold 0, idx 1406). Grad-CAM1D.

Fig. 3: Qualitative examples for the word task (Table XII). Left column: averaged decoder cross-attention heatmaps. Right column: Grad-CAM1D overlaid on input RMS. Rows correspond to increasing error severity over the nonzero errors ($e > 0$): $p50$, $p90$, and $p99$.

20.60). This pattern is practically meaningful: WER is typically the user-visible failure mode in word-level recognition, and the AR decoder appears to reduce the number of completely wrong words even if it introduces additional character-level edits within partially correct predictions.

A plausible interpretation is that the AR objective changes the model's inductive bias from alignment-marginalized frame labeling (CTC) to explicit conditional sequence modeling under teacher forcing. Under CTC, decoding relies on collapsing repeats and blanks, which can preserve character-level accuracy in aligned regions yet still yield whole-word failures when alignment is ambiguous or when local confusions compound. Under AR decoding, the model is trained to produce globally coherent sequences conditioned on both the encoder memory and a strong autoregressive context, which can reduce catastrophic word-level substitutions but may introduce substitution/insertion errors when termination (EOS) and length control are imperfect or when exposure bias interacts with greedy decoding.

A practical implication is that AR decoding should be evaluated and tuned with deployment-aligned objectives: beyond CER/WER, decoding strategies (beam search, EOS calibration, length normalization, and coverage penalties) are likely to matter more than under CTC be-

cause generation quality is inseparable from the decoding procedure. [11, 9, 1]

B. Batch-wise rectangularization: an enabling step with nontrivial consequences

A key technical contribution is the batch-wise rectangularization strategy: within each minibatch, encoder sequences are padded only to the batch maximum and masked appropriately, enabling efficient Transformer decoding without global fixed-length padding. [10, 5] Beyond engineering convenience, this design choice is consequential for optimization stability and compute: it preserves GPU efficiency while preventing systematic over-padding that would inflate attention cost and memory. At the same time, the decoder still attends over the full encoder memory, and the computational burden remains large relative to the original CTC baseline (e.g., word-level MACs increase from single-digit millions for Rewi-B to hundreds of millions for Transformer variants). [10] This motivates careful discussion of the accuracy–compute envelope: AR Transformers may be attractive when the application values improved WER and a unified seq2seq training/deployment pipeline, but CTC remains compelling for edge-constrained settings.

C. Why BPE tokenization did not improve performance

Across BPE100/200/300, there is no consistent improvement over character-level decoding on either task. This negative result is informative rather than disappointing. In this setting, (i) word targets are short and drawn from a constrained distribution (English/German words with case/diacritics), so the marginal benefit of subword-level language modeling is limited; (ii) BPE introduces token boundary decisions that may not align with kinematic evidence, creating new error modes where the model must infer subword segmentation from sensor traces; and (iii) training dynamics change meaningfully with token length (teacher forcing sequences become shorter but "more semantic"), often requiring retuning of schedules and regularization. [8] Given these considerations, character decoding appears to be a strong default for IMU HWR under the current architecture and training setup, and BPE may only become competitive if paired with stronger language constraints (e.g., explicit LM fusion, constrained decoding, or larger text corpora for pretraining).

D. Gated attention: consistent gains and a plausible mechanistic story

Both headwise and elementwise SDPA-output gating deliver clear improvements over the ungated AR Transformer, reducing both CER and WER across tasks (e.g., word: Ours-B-AR 12.80/20.60 $\rightarrow$ 10.29–10.37 / 17.43–17.54; sentence: 11.35/15.83 $\rightarrow$ 8.68/12.25–12.30). These gains are consistent with the intended role of gating: to modulate attention outputs and mitigate pathological attention behaviors (e.g., overly confident head outputs or unstable token-to-memory focus). :contentReference[6] While the project does not claim causal interpretability, the qualitative analysis aligns with this story: severe errors often coincide with unstable alignment patterns (fragmentation or collapse), suggesting that mechanisms which stabilize attention flow can translate into measurable improvements. [4]

A further practical point is that gating improves performance without materially changing parameter count or MACs (it is a lightweight modification), making it an attractive default augmentation for AR variants in this domain. [6]

E. Error distributions: bimodality, heavy tails, and what they imply

The distributional analysis (Table XI) shows a strong mass at exact matches ($e = 0$) and a heavy-tailed nonzero regime. Exact match rates are roughly $\sim$79% on both datasets, while nontrivial tail probabilities remain (e.g., $P(e > 0.5)$ is $\sim$12% for word and $\sim$17% for sentence; $P(e > 1)$ is $\sim$1% word and $\sim$4% sentence). This matters because average metrics can obscure operational risk: a model that is usually correct but occasionally fails catastrophically can be unacceptable in downstream human-in-the-loop workflows unless failures are detectable or mitigated.

The length correlation analysis supports the choice of normalized Levenshtein distance for cross-task comparability: raw edit distance correlates with label length, while normalized error shows weak length dependence overall and becomes slightly negative when conditioning on errors only, indicating that longer targets do not systematically degrade relative accuracy. From a modeling perspective, this suggests that remaining failures are less about "long sequence fatigue" and more about specific ambiguity modes (writer shift, token-time alignment breakdown, or decoding priors).

F. Collision behavior: decoder priors, frequency bias, and writer shift

The collision analysis diagnoses a specific failure mode: incorrect predictions whose output string exactly matches the ground-truth label of another sample. Collisions concentrate on frequent labels (median `gt_label_count` in the mid-teens with a substantial upper tail), and curated collision pairs show little systematic increase in IMU similarity relative to a within-fold random baseline. Together with the pronounced writer heterogeneity—a small subset of writers contributes a disproportionate share of collisions (lift $\approx$ 2–3 for top-K groups) and `collision_given_error` can exceed

50%—these results are most consistent with decoder-prior effects (frequency bias and weak input conditioning) modulated by writer-specific shift, rather than a dominant input-space confusion explanation.

Overall, collided predictions concentrate on frequent ground-truth strings (high `gt_label_count`), while curated collision pairs exhibit only marginal differences in IMU similarity compared to within-fold random baselines. In addition, collisions are not uniformly distributed across writers: a small subset of writers contributes a disproportionate share of collision samples, as reflected by elevated lift values and high `collision_given_error` among the most collision-prone writers.

As immediate follow-ups, we will (i) report the most frequent collided predictions together with their `gt_label_count` (globally and for collision-prone writers) to identify dominant collision modes, and (ii) evaluate mitigation strategies that target both axes: frequency-aware reweighting or balanced sampling, and stronger conditioning/alignment mechanisms (e.g., auxiliary alignment losses or decoding/conditioning interventions), before exploring more complex representation-level objectives such as contrastive learning.

G. Qualitative evidence: alignment stability as a useful diagnostic

The attention/Grad-CAM inspection provides interpretable evidence that complements the distributional findings. Across sampled regimes, increasing error severity coincides with less reliable token–time alignment in cross-attention: near-miss errors often retain partial structure but exhibit repeated vertical bands (reusing the same encoder positions across successive tokens), whereas severe failures show pronounced attention collapse onto a narrow encoder region. In contrast, Grad-CAM signals are less consistent—sometimes localized, sometimes broad—and can saturate in high-confidence predictions, which is a known limitation of gradient-based saliency methods. Consequently, cross-attention patterns appear to be the more reliable alignment indicator in this setup, while Grad-CAM is best treated as a supportive, non-calibrated diagnostic.

A practical implication is that these alignment pathologies can serve as error-detection cues: attention collapse or strong repetition could be used for confidence estimation or to trigger fallback strategies (e.g., rerun decoding with beam search, request a rewrite, or apply lexicon constraints). [9]

Given the fold-wise stability of the quantitative results and the collision findings (Section V-C), the next step is to connect qualitative signatures to recurrent failure categories more systematically. In the thesis, we will (i) scale inspection to quantile-selected samples spanning the nonzero error spectrum ($p50$/$p90$/$p99$ over $e > 0$), (ii) contrast collision vs. non-collision errors to test whether collisions exhibit distinctive attention/saliency signatures, and (iii) incorporate writer-stratified qualitative checks to assess whether writer-specific shift corresponds to systematic changes in alignment behavior.

H. Concatenation augmentation: why it failed and what it suggests

Overall, concatenation-based augmentation was detrimental in the current setup. A plausible explanation is that concatenation changes the data-generating process in multiple ways that are not well matched by the model and decoding procedure: (i) very long encoder memories increase optimization difficulty and amplify cross-attention instability; (ii) the absence of an explicit separator token forces the decoder to infer boundaries purely from dynamics, which may be ambiguous even within a single writer; and (iii) concatenation interacts with per-sample augmentations (noise, drift, warping), potentially creating composite sequences that are less physically plausible than naturally continuous handwriting. These findings suggest that long-context robustness should be pursued with more structured approaches, such as inserting explicit boundary markers, using curriculum schedules over K and `max_T`, or adopting architectures that handle long sequences more gracefully (e.g., chunked attention or hierarchical encoders), rather than naive end-to-end concatenation.

I. Limitations and threats to validity

Several limitations should be acknowledged to frame the conclusions appropriately:

- **Decoding regime:** AR results are reported under greedy decoding. Given the sensitivity of AR models to decoding, beam search and EOS calibration may change the CER/WER trade-off materially, and should be considered before final conclusions about objective superiority. [11, 9]
- **Tokenization scope:** BPE results are limited to three merge-operation settings without extensive hyperparameter retuning. While the observed lack of gains is meaningful, a stronger BPE evaluation would include schedule/regularization sweeps and possibly LM fusion.[8]
- **Qualitative sample size:** The attention/Grad-CAM analysis is illustrative and intentionally not exhaustive; scaling the inspection to quantile-selected samples is necessary to avoid over-interpreting a small set of examples. [4, 7]

J. Actionable next steps

Based on the evidence in this report, the most promising immediate extensions are:

1) **Decoding improvements:** introduce beam search, length normalization, EOS calibration, and (optionally) coverage penalties to reduce attention collapse and catastrophic tails.
2) **Bias mitigation for collisions:** evaluate label-frequency reweighting or balanced sampling; analyze top collided labels globally and per collision-prone writers to separate global priors from writer-driven effects.
3) **Hybrid objectives:** explore CTC+AR multitask training or alignment-aware AR objectives (e.g., RNN-T-style ideas) to retain CTC's alignment robustness while benefiting from AR conditional modeling. [2]
4) **Writer robustness:** incorporate writer-stratified evaluation routinely and test training interventions targeted at writer shift (balanced batches, writer-conditional normalization, or contrastive objectives on encoder embeddings).

a) Overall takeaway.: The project demonstrates that an IMU-HWR system can be moved from a CTC pipeline to an attention-based autoregressive decoder without prohibitive engineering complexity by using batch-wise rectangularization, and that lightweight gating at the attention output provides consistent gains. The remaining challenges are concentrated in the heavy tail—catastrophic decoding and collision-prone regimes—where decoding strategy, conditioning strength, and writer robustness are likely to deliver the next tranche of improvements.

VII. Conclusion

This project developed and evaluated an attention-based autoregressive (AR) encoder–decoder for IMU-based handwriting recognition as a step toward a thesis on robust sequence modeling and failure analysis in sensor-driven HWR. Starting from a competitive CTC baseline, we implemented a Transformer decoder pipeline that can be trained and evaluated efficiently on highly variable-length sequences through batch-wise rectangularization and masking. This design choice proved essential for making AR decoding practical under the compute and memory constraints imposed by long IMU traces.

Across word- and sentence-level validation sets, AR decoding achieved markedly stronger word-level recognition in terms of WER while exhibiting a distinct CER–WER trade-off relative to CTC. In addition, gating the SDPA output (both headwise and elementwise variants) consistently improved performance across tasks, providing an inexpensive and effective mechanism to stabilize attention-mediated information flow during decoding. By contrast, BPE tokenization did not deliver systematic gains over character-level decoding under the current data regime and training configuration, suggesting that subword modeling is not the primary bottleneck for IMU-HWR in this setting.

A key contribution of this work is moving beyond aggregate metrics to characterize error structure. Normalized edit-distance analysis revealed strongly bimodal outcomes—many exact matches alongside a heavy-tailed nonzero error regime—indicating that average performance can mask operationally important tail risk. Collision analysis further showed that some incorrect predictions correspond to frequent labels, and writer-stratified results suggested that collision propensity is concentrated in a subset of writers, consistent with an interaction between label-frequency bias and writer-specific distribution shift. Qualitative inspection using decoder cross-attention and encoder-side Grad-CAM1D provided interpretable evidence that higher-severity failures are often associated with degraded token–time alignment, including fragmented focus and, in extreme cases, attention collapse onto narrow encoder regions.

In summary, the project establishes a robust experimental and diagnostic foundation for the thesis: a practical AR Transformer pipeline for IMU-HWR, a lightweight attention-gating improvement with consistent gains, and a multi-level analysis framework (distributional, collision-based, writer-stratified, and qualitative) that isolates where and how the system fails. The most promising next steps are to reduce heavy-tail failures via improved decoding (e.g., beam search and EOS/length calibration), to formalize collision-aware evaluation and mitigation strategies, and to strengthen robustness to writer shift through targeted training interventions and broader, quantile-driven qualitative analysis.

References

[1] Samy Bengio et al. "Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2015.

[2] Alex Graves. "Sequence Transduction with Recurrent Neural Networks". In: *arXiv preprint arXiv:1211.3711* (2012).

[3] Alex Graves et al. "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks". In: *Proceedings of the 23rd International Conference on Machine Learning (ICML)*. 2006, pp. 369–376. DOI: 10.1145/1143844.1143891.

[4] Sarthak Jain and Byron C. Wallace. "Attention is not Explanation". In: *Proceedings of NAACL-HLT*. 2019, pp. 3543–3556. DOI: 10.18653/v1/N19-1357.

[5] Jindong Li et al. “Robust and Efficient Writer-Independent IMU-Based Handwriting Recognition”. In: *arXiv preprint arXiv:2502.20954* (2025).

[6] Zhen Qiu et al. “Gated Attention for Large Language Models: Non-linearity, Sparsity, and Attention-Sink-Free”. In: *arXiv preprint arXiv:2505.06708* (2025).

[7] Ramprasaath R. Selvaraju et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 618–626. DOI: 10.1109/ICCV.2017.74.

[8] Rico Sennrich, Barry Haddow, and Alexandra Birch. “Neural Machine Translation of Rare Words with Subword Units”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2016, pp. 1715–1725. DOI: 10.18653/v1/P16-1162.

[9] Zhaopeng Tu et al. “Modeling Coverage for Neural Machine Translation”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*. 2016, pp. 76–85. DOI: 10.18653/v1/P16-1008.

[10] Ashish Vaswani et al. “Attention Is All You Need”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2017.

[11] Yonghui Wu et al. “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation”. In: *arXiv preprint arXiv:1609.08144* (2016).