A dissertation submitted in partial fulfilment of the
requirements for the degree of

Master of Technology in Computer Science

Air-Writing Recognition

Author:
Gaurang Shukla
Roll No.: **CS2415**

Supervisor:
Umapada Pal
Computer Vision and Pattern
Recognition Unit (CVPRU)

Indian Statistical Institute, Kolkata

June 2026

Certificate

This is to certify that the dissertation entitled **"Air-Writing Recognition"** submitted by **Gaurang Shukla** (Roll No. **CS2415**) in partial fulfilment of the requirements for the degree of *Master of Technology in Computer Science* at the Indian Statistical Institute, Kolkata, is a bonafide record of work carried out under my supervision. The contents of this dissertation, in full or in part, have not been submitted to any other institute or university for the award of any degree or diploma.

Umapada Pal 10/06/2026

Umapada Pal
Supervisor
Computer Vision and Pattern Recognition Unit
Indian Statistical Institute, Kolkata

Declaration of Authorship

I, **Gaurang Shukla**, declare that this dissertation titled "*Air-Writing Recognition*" and the work presented in it are my own. I confirm that:

- This work was carried out wholly while in candidature for the degree of Master of Technology in Computer Science at the Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata, under the supervision of **Umapada Pal**.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given; with the exception of such quotations, this dissertation is entirely my own work.
- I have acknowledged all main sources of help.
- All third-party code, datasets, and pre-trained models used in this work, in particular the WiTA dataset, MediaPipe, and PyTorch, are cited appropriately.
- The experimental results are reported honestly, with every quantitative claim traceable to a concrete experimental artifact.

Gaurang 10/06/2026

Gaurang Shukla
Roll No. CS2415

Acknowledgements

I am grateful to my supervisor, **Umapada Pal**, for the guidance, patience, and critical feedback that shaped this work, and for encouraging an honest, hypothesis-driven approach in which negative results are treated as evidence rather than failures. I thank the faculty and staff of the Indian Statistical Institute for providing an excellent research environment. I acknowledge the authors of the WiTA dataset for making the data publicly available, and the open-source communities behind PyTorch, MediaPipe, and the broader scientific-Python ecosystem. Finally, I thank my family and friends for their constant support.

Abstract

Air-writing is the act of tracing characters or words in free space with a fingertip, recorded by a camera, giving a touch-free input modality for smart displays, augmented and virtual reality, and assistive interfaces. It is difficult because the finger never lifts: connecting strokes join adjacent letters with no pen-up signal to mark boundaries, and the same word varies widely in scale, position, and slant across writers. The WiTA benchmark of Kim et al. provides a large, person-disjoint dataset and a baseline that treats each clip as RGB video, recognised by a spatio-temporal 3D residual network trained with a CTC objective, reaching a character error rate (CER) of 0.292 on the English subset. The main goal of this dissertation was to improve on this error rate, which we achieve: we replace raw video with an explicit fingertip-trajectory sequence extracted from hand landmarks, fed to a Conformer encoder with a joint CTC/attention head. The resulting system attains a test CER of 0.219, improving on the published 0.292 of Kim et al. and 0.299 of Tan et al. by 15–27% relative.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Air-writing recognition

Air-writing is the act of tracing characters or words in three-dimensional free space with a fingertip or hand, while a camera or other sensor records the path and turns it into text. Because it needs no surface and almost no hardware, it fits situations where touching a screen is awkward or unwelcome: smart displays, virtual and augmented reality, controls in a car or an operating theatre, and assistive interfaces.

Why it is hard becomes clearer by comparison. Writing on a tablet is *online* handwriting; writing on paper and scanning it is *offline* handwriting, or OCR. Air-writing differs from both in two ways. There is no pen-up or pen-down signal, so the finger is always "writing": each letter runs straight into the next through a connecting stroke, and nothing marks where one character ends and the following one starts. The writing is also unconstrained. With no surface to rest against, people differ widely in how large they write, how much they slant, how fast they move and where in the frame they work, and they often retrace new letters on top of old ones rather than moving tidily from left to right. What the camera records is a single tangled stroke whose division into letters is genuinely ambiguous if one looks only at its shape (Figure 1.1).

1.2 The WiTA benchmark

WiTA, short for *Writing in The Air* [1], is a large public benchmark for the task. Its subjects write English and Korean words in the air in front of a single third-person camera, and every clip is a short RGB video of one word. The English part splits in two: a *lexical* subset, taken from a fixed list of common words, and a *non-lexical* subset of random character strings that spell nothing. Results are given as character

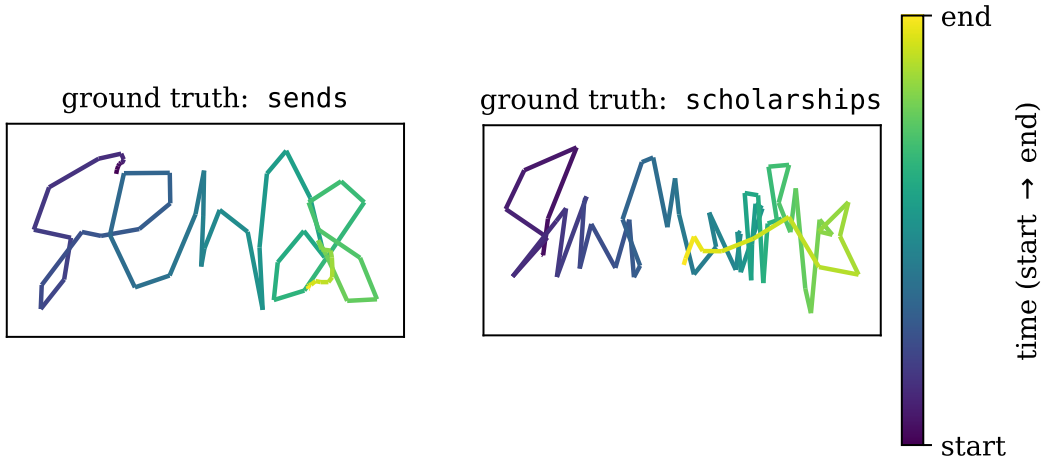

Figure 1.1: Index-fingertip trajectories of two real WiTA test clips, with colour encoding time. Because the finger never lifts, each word is one continuous, self-overlapping stroke whose letters cannot be read off the shape.

error rate, listed separately for the two subsets, because they test different things. A lexical word can be guessed in part from a language model; a random string offers no such help.

Kim et al. [1] treat each clip as a video and run a spatio-temporal 3D residual network (ST-R3D) over it, trained with a Connectionist Temporal Classification (CTC) loss. On the person-disjoint English split this reaches an overall test CER of 0.2924, with 0.2810 on lexical words and 0.3646 on non-lexical strings. One detail of their setup matters here: the decoder is CTC only, and they list attention-based decoding as future work. Part of what this dissertation does, adding an attention head on top of CTC, is therefore a direct attempt at that extension. Their 0.2924 is the number this work sets out to explain and then to beat.

1.3 Motivation

The motivation of this dissertation is to improve recognition accuracy on the WiTA air-writing benchmark. The published baseline of Kim et al. reaches an overall English test CER of 0.2924, and our aim is to lower this error rate. Beyond the benchmark figure, accurate air-writing recognition has practical value: it offers a touch-free text-entry modality for smart displays, augmented and virtual reality, and assistive interfaces, where touching a screen is awkward or impractical. A further attraction is that a model reading an explicit fingertip trajectory rather than raw video is small and far easier to interpret and deploy than a 3D CNN.

1.4 Contributions

This dissertation makes the following contributions.

1. **A better input representation:** in place of raw RGB video, an explicit fingertip-trajectory sequence, comprising position, velocity, and acceleration extracted from hand landmarks at a sufficient temporal resolution (T=64), giving a compact and interpretable input for unconstrained air-writing (Chapter 4).

2. **A better recognition architecture:** a Conformer encoder with a joint CTC/attention head, which adds the attention-based decoding that Kim et al. left as future work while remaining small, with a few million parameters operating on a 190-dimensional landmark sequence rather than on video.

3. **Better results at lower complexity:** in combination, the proposed representation and architecture attain a test CER of 0.219, improving on both published RGB-video systems, 0.292 for Kim et al. [1] and 0.299 for Tan et al. [7], by 15–27% relative, while being substantially smaller and easier to deploy than a 3D CNN.

1.5 Organisation

Chapter 2 reviews the relevant background and related work. Chapter 3 describes the dataset, task, evaluation metric, and an audit of the data. Chapter 4 presents the methodology of the final system: landmark extraction, the trajectory features, per-clip normalisation, temporal sampling, and the Conformer CTC/attention model. Chapter 5 reports the experiments and results, and Chapter 6 concludes.

Chapter 2

Background and Related Work

Part I Background

2.1 Air-writing and online handwriting recognition

Air-writing recognition sits between gesture recognition and online handwriting recognition (OHR). The shared premise is that handwriting is fundamentally a *temporal* signal: the order in which strokes are produced disambiguates shapes that overlap in space. Air-writing inherits this property but removes the pen-state signal, which makes the temporal cue even more important. This chapter first reviews the technical components the dissertation builds on (Part I) and then surveys prior air-writing research and the related literature (Part II), before stating the research gap that motivates the work.

2.2 Challenges of air-writing recognition

Air-writing presents two structural difficulties relative to tablet or paper handwriting. First, there is no pen-up/pen-down signal, so successive letters are joined by connecting strokes with no cue marking where one character ends and the next begins. Second, the writing is unconstrained: with no surface to anchor against, writers vary widely in size, slant, speed, and frame position and frequently overwrite earlier letters. The recognised signal is therefore a single, self-overlapping trajectory whose segmentation is ambiguous from spatial appearance alone and whose position and scale vary across writers.

2.3 Sequence recognition and temporal modelling

These properties make air-writing a sequence-labelling problem: a variable-length input must be mapped to a variable-length character string with no given alignment between the two. Two ingredients are accordingly central: an encoder that models temporal context along the input, and a training objective that copes with the unknown input–output alignment. The rest of Part I introduces the CTC and attention objectives and their hybrid, on which the proposed recogniser is built.

2.4 Connectionist Temporal Classification

Connectionist Temporal Classification (CTC) [2] enables training a sequence model without a frame-level alignment between the input of length T and the target label sequence $\mathbf{y}$ of length $U \leq T$. A special *blank* symbol $\varnothing$ is added to the alphabet. Given a per-frame distribution $p(k \mid \mathbf{x})_t$ over the extended alphabet, a path $\boldsymbol{\pi} \in \{\mathcal{A} \cup \{\varnothing\}\}^T$ has probability

$$p(\boldsymbol{\pi} \mid \mathbf{x}) = \prod_{t=1}^{T} p(\pi_t \mid \mathbf{x})_t. \tag{2.1}$$

A many-to-one map $\mathcal{B}$ collapses repeated labels and removes blanks (e.g. $\mathcal{B}(a\varnothing abb) = aab$). The probability of the target is the sum over all paths that collapse to it,

$$p(\mathbf{y} \mid \mathbf{x}) = \sum_{\boldsymbol{\pi} \in \mathcal{B}^{-1}(\mathbf{y})} p(\boldsymbol{\pi} \mid \mathbf{x}), \qquad \mathcal{L}_{\text{CTC}} = -\log p(\mathbf{y} \mid \mathbf{x}), \tag{2.2}$$

which is computed efficiently by the standard forward–backward (dynamic-programming) algorithm in $O(TU)$ time. CTC is well suited to air-writing because it requires no manual segmentation of the continuous trajectory into characters.

2.5 Attention-based sequence recognition and joint CTC/attention

An attention-based decoder [3] models the label sequence autoregressively,

$$p(\mathbf{y} \mid \mathbf{x}) = \prod_{u=1}^{U} p(y_u \mid y_{<u}, \mathbf{x}), \qquad \mathcal{L}_{\text{attn}} = -\sum_{u=1}^{U} \log p(y_u \mid y_{<u}, \mathbf{x}), \tag{2.3}$$

attending over the encoder memory at each output step. Attention decoders learn an implicit language model over the output alphabet, which is helpful for real words

but can be harmful for random strings. The hybrid CTC/attention framework [4] combines the two with a single weight λ:

$$\mathcal{L} = \lambda \mathcal{L}_{\text{CTC}} + (1 - \lambda) \mathcal{L}_{\text{attn}}. \tag{2.4}$$

CTC provides monotonic alignment and a strong conditional-independence regulariser, while attention contributes label-context modelling; the two heads share an encoder and are trained jointly. The choice of λ and the training schedule are given in Chapter 4.

Part II Related Work

The preceding sections covered the technical machinery this dissertation builds on. This part surveys prior air-writing research and the most relevant adjacent literature, positioning the present work and exposing the gap it addresses.

2.6 Sensor-based air-writing systems

The earliest air-writing systems recovered the writing trajectory directly from active or wearable sensors, side-stepping visual tracking. Inertial measurement units worn on the hand or embedded in a stylus capture acceleration and angular velocity [8, 9]; controller- or depth-based devices such as the Leap Motion and Microsoft Kinect supply 3D hand or fingertip coordinates [10, 11, 12]; and more recent work senses the motion with radar or WiFi [13, 14]. Early recognisers handled isolated digits or characters with dynamic time warping or hidden Markov models [15], later superseded by recurrent networks [23]. Such systems are accurate in instrumented, single-user settings, but depend on specialised hardware and do not transfer to ordinary cameras.

2.7 Vision-based air-writing systems

Camera-based systems recognise writing from ordinary RGB (and occasionally depth) video, removing the need for worn hardware. They must first localise and track the hand or fingertip and then transcribe the recovered path, which introduces sensitivity to background clutter, viewpoint, lighting, and tracking failure [16, 17, 18]. The WiTA benchmark used in this dissertation belongs to this camera-based family. Because the trajectory is *inferred* rather than measured, how it is represented (raw video, a rendered

image, or an explicit landmark sequence) becomes a central design decision, which this dissertation studies directly.

2.8 Online handwriting recognition relevant to air-writing

Air-writing is closely related to online handwriting recognition (OHR), which transcribes pen-tip trajectories captured on a tablet or stylus. Mature OHR pipelines model the trajectory as a temporal sequence and train with CTC, with bidirectional LSTM networks a long-standing strong baseline for unconstrained handwriting [19], including online character recognition [20]. Air-writing is essentially OHR without a pen-up/pen-down signal: the same sequence framing applies, but the absence of segmentation cues makes the temporal ordering of the trajectory even more important. This connection motivates the trajectory-sequence formulation adopted later in this dissertation.

2.9 Deep-learning approaches for air-writing

Recent air-writing and gesture systems are predominantly deep sequence models: temporal CNNs, LSTM and GRU networks, Transformers, and, for video input, 3D convolutional networks, frequently paired with CTC for unsegmented word- or sentence-level recognition [21, 22, 23, 24]. These works have steadily improved accuracy, but they concentrate on the recognition architecture and generally treat the input representation as fixed pre-processing.

2.10 Landmark-based recognition methods

A complementary line of work replaces raw pixels with extracted skeletons. Hand and body landmarks from pose estimators such as MediaPipe Hands [6] are fed to sequence models or to graph-convolutional networks for skeleton-based action recognition [25]. Landmark representations are compact and largely invariant to appearance, which makes them attractive for air-writing; however, *raw* landmark coordinates remain tied to the writer's absolute position and scale in the frame, a limitation this dissertation identifies as decisive.

2.11 The WiTA benchmark and previous state of the art

The WiTA (Writing in The Air) benchmark [1] is the largest public air-writing dataset and the direct point of comparison for this work. It contains English and Korean words written in the air before a single camera, divided into lexical and non-lexical subsets and evaluated under a person-disjoint split (Chapter 3). The reference system treats each clip as video and applies a spatio-temporal 3D residual network (ST-R3D) trained with a CTC objective, reaching an overall English test CER of 0.2924 (0.2810 lexical, 0.3646 non-lexical); attention-based decoding is identified as future work. Subsequently, Tan et al. [7] reported a Transformer-based system (TR-AWR) on the same benchmark, reaching an overall English CER of 0.2986. Both published systems are compared against in Chapter 5; the ST-R3D CTC result of Kim et al. [1] remains the principal baseline this dissertation seeks to understand and improve.

2.12 Research Gap and Motivation

A clear gap stands out across this literature: prior work is overwhelmingly concerned with the recognition *architecture* (which encoder, decoder, and loss) while treating the input representation and its temporal sampling as fixed pre-processing. The effect of how the fingertip trajectory is represented and how finely it is sampled in time has not, to the best of our knowledge, been studied systematically for air-writing.

This dissertation addresses that gap. Rather than proposing another architecture, it holds the recognition model essentially fixed and varies the representation of the fingertip trajectory and its temporal resolution under a controlled, single-variable ablation, on the same person-disjoint WiTA split and a one-shot test protocol. It thereby (i) isolates the effect of per-clip landmark normalisation and of increased temporal resolution, and (ii) establishes which *representational* choices, rather than which architecture, govern recognition accuracy. The remainder of the dissertation develops and evaluates this argument.

Chapter 3

Dataset, Task, and Evaluation

3.1 The WiTA English dataset

We use the English portion of WiTA under the paper's person-disjoint 8 : 1 : 1 split, so that no writer appears in more than one of the train, validation, and test partitions. Each clip is a short RGB video of one written word, labelled with its character string and tagged as lexical (`freq_word`) or non-lexical. Directory names encode the signer, demographic attributes, language, and subset type.

The full benchmark is large and was collected under deliberately varied conditions: 122 participants each wrote 195 phrases across indoor and outdoor environments, yielding 209,926 video instances in total [1]. Table 3.1 summarises the principal statistics of the dataset and of the English portion used in this dissertation, as reported by Kim et al. [1].

Two of these statistics bear on the representational interventions of this dissertation. First, the writing *scale* varies widely across writers: the horizontal scale ranges from 7.3 to 52.8 pixels, roughly a factor of seven. This writer-dependent variation motivated per-clip normalisation (Section 4.4); the controlled experiment of §5.2.2, however, finds that normalising it away does not improve accuracy. Second, the clip lengths motivate the temporal resolution: English lexical clips contain on average 78.75 native frames, so a fixed sampling at T=32 discards more than half of the available temporal detail, whereas T=64 retains most of it (Section 4.5).

3.2 Data audit

Before modelling, we audited the data to rule out trivially large gains and to characterise quality. The audit (Table 3.2) confirmed that the English data is **single-view**

Table 3.1: Statistics of the WiTA dataset and its English portion, as reported by Kim et al. [1]. Frame and character counts are given as average / standard deviation; writing-scale figures are in pixels.

Property	Value
Benchmark (both languages)	
Participants	122
Total video instances	209,926
Languages / sub-datasets	2 (English, Korean) / 5
Capture	RGB, 29 fps, 224×224, third-person
Environments	indoor and outdoor (varied backgrounds)
Split	person-disjoint 8 : 1 : 1 (train/val/test)
English lexical	
Vocabulary source	top-6000 Google 1B words
Word length (characters)	6.59/2.54
Clip length (frames)	78.75/28.65
English non-lexical	
Generation	random strings, length 3–7
Word length (characters)	5.03/1.41
Clip length (frames)	68.08/21.49
English writing behaviour	
Writing speed	3.57 characters/s
Writing scale, x (px)	18.35/7.27 (range 7.3–52.8)
Writing scale, y (px)	14.39/8.64 (range 3.7–57.5)

Table 3.2: Data audit of the WiTA English split (clips and unique signers).

Split	Lexical clips	Non-lexical clips	Signers
Train	7015	1410	89
Val	900	180	12
Test	900	180	12

(the trailing directory token is the subset type, not a camera angle, so there is no extra multi-view data to exploit), and that the landmark cache already covers 96.8–100% of available clips. The lexical vocabulary is a fixed set of 5,762 unique words; only $\approx 51\%$ of test lexical words also occur in the training lexical vocabulary, which (as we show later) makes the task genuinely open-vocabulary and rules out a closed-dictionary shortcut.

3.3 Tracking quality

Hand detection by MediaPipe is generally reliable (per-clip detection rate ≈ 0.9–1.0 for most signers), but a few signers track poorly (Figure 3.2); the worst, `WBS`, has a

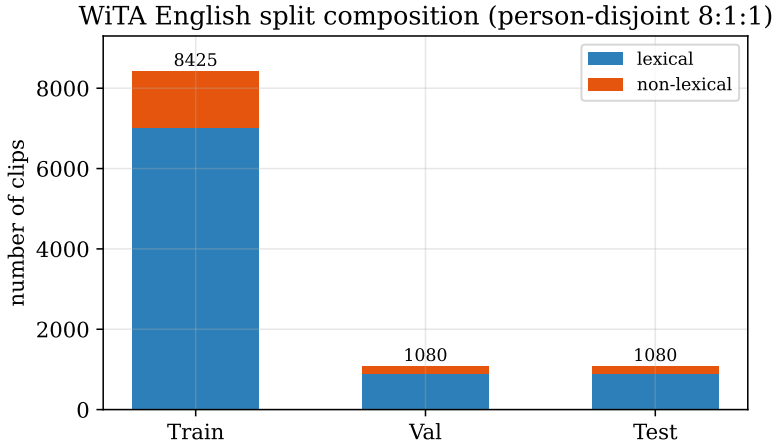

Figure 3.1: Composition of the WiTA English person-disjoint split by partition and subset (counts from Table 3.2). Lexical clips dominate ($\approx 83\%$ of the 10,585 English clips), and the split is writer-disjoint, so generalisation to unseen signers is tested directly.

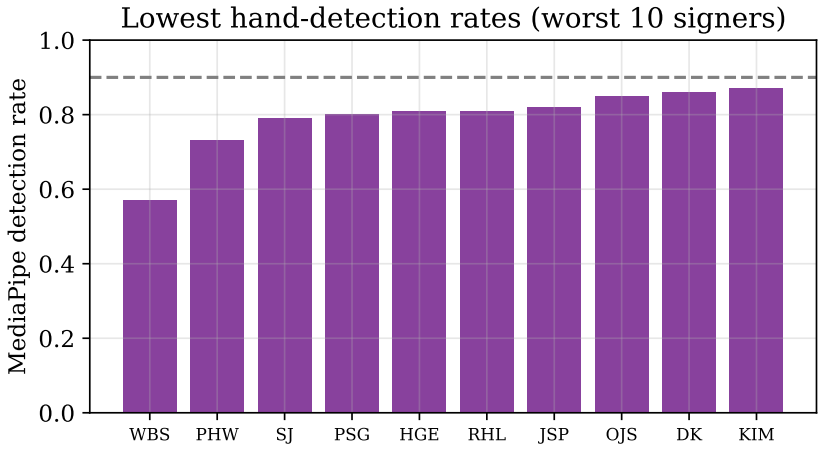

Figure 3.2: MediaPipe hand-detection rate for the ten worst signers. Most signers exceed 0.9; a small tail tracks poorly and accounts for much of the residual error.

detection rate of only 0.57. As the per-signer analysis of Chapter 5 shows, residual error concentrates on poorly tracked signers, which indicates that tracking quality, not the recogniser, is the remaining bottleneck for the hardest cases.

3.4 Evaluation metric

Performance is measured by **character error rate** (CER), the length-normalised Levenshtein (edit) distance between the predicted string $\hat{\mathbf{y}}$ and the ground truth $\mathbf{y}$:

$$\mathrm{CER} = \frac{\sum_i \mathrm{Lev}(\hat{\mathbf{y}}_i, \mathbf{y}_i)}{\sum_i |\mathbf{y}_i|} = \frac{\sum_i (S_i + D_i + I_i)}{\sum_i N_i}, \tag{3.1}$$

where S, D, I are substitutions, deletions, and insertions and N is the reference length. We report CER separately for the lexical and non-lexical subsets and overall, matching [1].

3.5 Evaluation discipline

A recurring risk in iterative work is implicitly tuning to the test set. We adopt two safeguards. (i) Model checkpoints are selected by validation CER only; the test set is evaluated *once* per reported system, gated by a marker file. (ii) Decoder choices (e.g. which head to use, language-model weights) are fixed on validation before the single test evaluation. As Chapter 5 reports, this discipline also surfaced a subtle *oracle* bug in the evaluation code, which we corrected before stating any comparison to the baseline.

Chapter 4

Methodology

This chapter specifies the proposed air-writing recognition system in full, in the order in which data flow through it, from the input clip to a recognised character string. The system is purely landmark-based: each clip is reduced to a sequence of hand-landmark coordinates, converted into kinematic features, normalised per clip, and transcribed by a Conformer encoder with a joint CTC/attention head. The empirical comparison of this design against alternative representations, and the analysis that motivated its choices, are reported in Chapter 5.

4.1 Overview of the proposed system

A guiding principle of the design is that the recogniser is held deliberately conventional: the contribution of this work lies in the *representation* the model consumes, not in the network itself. The recognition pipeline (Figure 4.1) comprises five stages. First, a hand-landmark extractor converts each frame of a clip into 21 hand landmarks. Second, the landmark stream is resampled to a fixed temporal length and augmented with velocity and acceleration, yielding a 190-dimensional per-frame feature vector. Third, each clip is normalised using its own spatial statistics, removing the writer's absolute position and scale. Fourth, a Conformer encoder maps the feature sequence to a sequence of contextualised states. Fifth, a joint CTC/attention head transcribes these states into a character string. The two heads share a single encoder and are trained under a joint objective (Section 4.6). The model is trained end-to-end with this hybrid objective and evaluated with a single, validation-selected decoder. The remaining sections specify each stage in sufficient detail to reproduce the system.

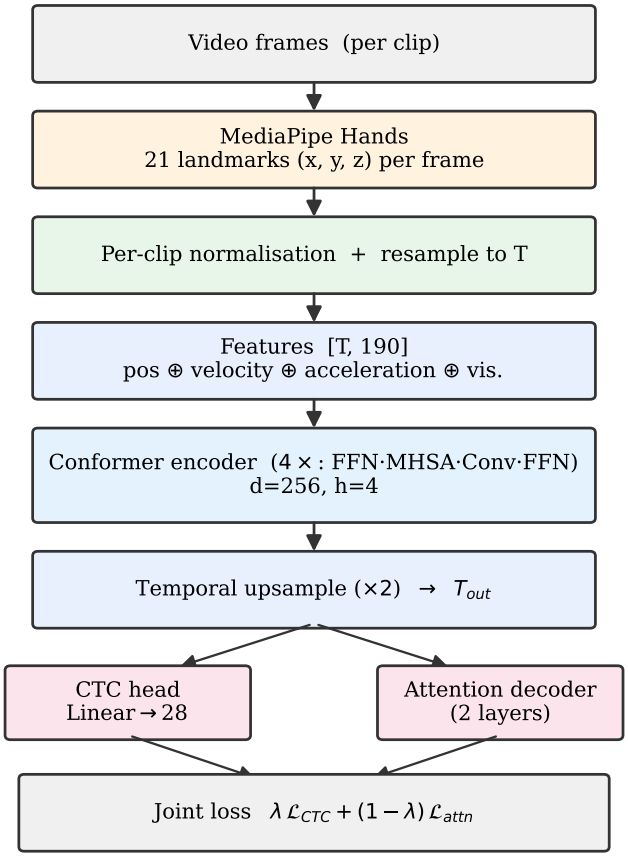

Figure 4.1: The recognition pipeline: MediaPipe landmarks → per-clip normalisation → position/velocity/acceleration features → Conformer encoder → joint CTC and attention heads, trained with the hybrid loss.

4.2 Landmark extraction

Hand landmarks are extracted with MediaPipe Hands [6], a real-time hand-tracking pipeline. For every frame the detector returns 21 hand landmarks, each expressed as a normalised coordinate (x, y, z), where $x, y \in [0, 1]$ are relative to the frame width and height and z is a relative depth. The index fingertip corresponds to landmark 8 and traces the written path; all 21 landmarks are retained so that the model has access to the full hand configuration.

Extraction proceeds frame by frame over each clip, with detection and tracking confidence thresholds of 0.5. When a hand is detected, its 21 landmarks are stored together with a visibility flag of 1. When no hand is detected (for example under motion blur or partial occlusion), the most recent valid landmark set is carried forward and the visibility flag is set to 0, so that downstream features remain well-defined while recording that the frame was imputed. The result, for each clip, is a variable-length

sequence of 21×3 landmark coordinates together with a visibility sequence. To avoid repeating this comparatively expensive step, landmark sequences are extracted once and cached to disk for reuse.

4.3 Feature representation

The landmark sequence is converted into a kinematic feature sequence. After temporal resampling to a fixed length T (Section 4.5), each frame t is described by the joint positions and their first and second temporal differences,

$$\mathbf{v}_t = \mathbf{x}_t - \mathbf{x}_{t-1}, \qquad \mathbf{a}_t = \mathbf{v}_t - \mathbf{v}_{t-1}, \tag{4.1}$$

where $\mathbf{x}_t \in \mathbb{R}^{21\times 3}$ stacks the 21 landmark coordinates of frame t, and $\mathbf{v}_t$ and $\mathbf{a}_t$ are the corresponding velocity and acceleration. Positions, velocities, and accelerations are flattened and concatenated with the per-frame visibility flag m_t, giving a 190-dimensional feature vector,

$$\mathbf{f}_t = \Big[\, \mathbf{x}_t \,\|\, \mathbf{v}_t \,\|\, \mathbf{a}_t \,\|\, m_t \,\Big] \in \mathbb{R}^{190}, \qquad 190 = 3 \times (21 \times 3) + 1. \tag{4.2}$$

Providing velocity and acceleration explicitly makes the writing dynamics directly available to the model rather than requiring it to infer them from positions. This is valuable for air-writing, where overlapping strokes render absolute position weakly discriminative and the motion itself carries letter identity.

Figure 4.2 summarises this construction. Every channel derives from the landmark coordinates and their motion, so the representation carries no appearance information.

4.4 Per-clip normalisation

Raw landmark coordinates are absolute positions within the camera frame and therefore depend on where in the frame a writer works and how large they write. To remove this dependence, each clip is normalised using only its own statistics. Let $\boldsymbol{\mu}$ be the mean landmark coordinate and σ the average per-axis standard deviation over the clip; every coordinate is centred and scaled,

$$\tilde{\mathbf{x}}_t = \frac{\mathbf{x}_t - \boldsymbol{\mu}}{\sigma}, \qquad \boldsymbol{\mu} = \frac{1}{T}\sum_t \mathbf{x}_t, \quad \sigma = \overline{\mathrm{std}_t(\mathbf{x}_t)}. \tag{4.3}$$

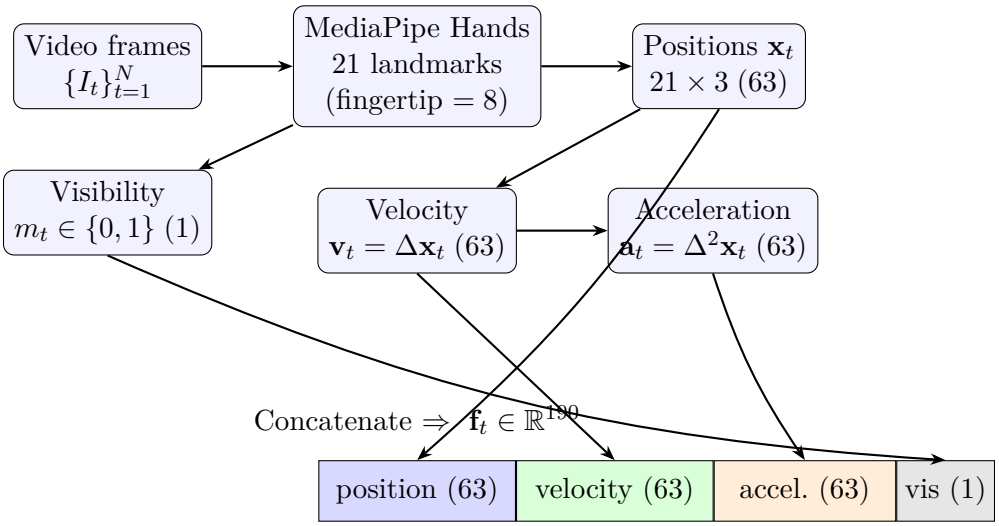

Figure 4.2: Construction of the per-frame landmark feature vector. For each frame, MediaPipe Hands yields 21 hand landmarks with normalised (x, y, z) coordinates (the index fingertip is landmark 8). Stacking the landmark coordinates forms the position block $\mathbf{x}_t$; its first and second temporal differences give velocity $\mathbf{v}_t$ and acceleration $\mathbf{a}_t$ (Equation 4.1), and a per-frame visibility flag m_t records whether a hand was detected. Concatenating the three 63-dimensional kinematic blocks with the visibility flag yields the 190-dimensional feature vector $\mathbf{f}_t$ of Equation 4.2.

Both statistics are computed from the clip alone, with no cross-clip or cross-split quantities, so the procedure introduces no information leakage and is applied identically to the training, validation, and test splits. Because velocity and acceleration are differences of position, the same scale factor σ is applied to them and the relation $\mathbf{v} = \Delta\mathbf{x}$ is preserved; centring leaves differences unchanged. The relative depth z and the visibility flag are left unmodified. The procedure makes the representation invariant to translation and scale: the same word written in different regions of the frame, or at different sizes, maps to the same normalised trajectory (Figure 4.3).

4.5 Temporal sampling

Landmark sequences differ in length from clip to clip and must be brought to a common length T for batched training. Each sequence is resampled to T frames by linear interpolation along the time axis, applied independently to the landmark coordinates and to the visibility sequence. The system uses a native temporal length of $T = 64$, obtained by extracting up to 128 frames per clip and resampling to 64.

The temporal length is treated as a design parameter rather than an implementation default. Air-written words vary in length, and longer words require more frames to represent the fine structure of individual letters and the connecting strokes between

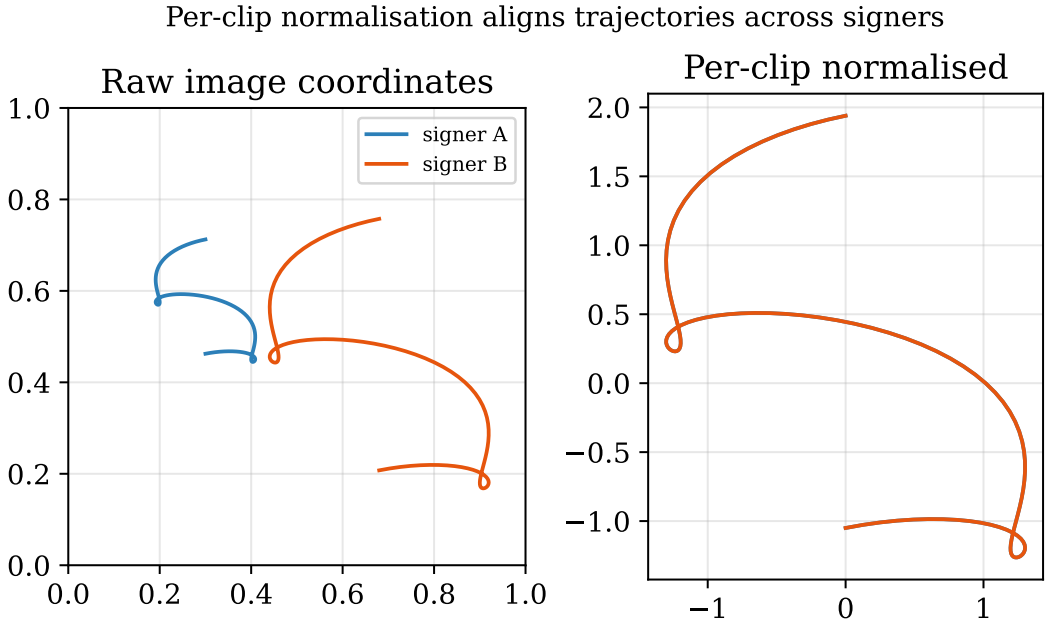

Figure 4.3: Per-clip normalisation (Equation 4.3). *Left*: two writers' trajectories in raw image coordinates differ in position and scale. *Right*: after centring and scaling, the shapes align, making the representation translation- and scale-invariant across signers.

them. As air-writing carries no pen-up signal, this temporal detail is what distinguishes one letter from the next, and coarse sampling aliases it away. A native length of $T = 64$ supplies sufficient temporal resolution for the longest words in the dataset while keeping the sequence short enough for efficient training.

4.6 Conformer-based recognition model

The recogniser is a Conformer encoder [5] with a joint CTC/attention head (Figure 4.1). It maps the normalised feature sequence $\tilde{\mathbf{F}} \in \mathbb{R}^{T\times 190}$ to a character string. Figure 4.4 shows the complete architecture with the tensor shape at each stage; the components are described in the subsections that follow. A single encoder is shared by the CTC head and the attention decoder, and the two branch losses are combined in the joint objective of Equation 2.4.

4.6.1 Input projection

The per-frame feature vector $\mathbf{f}_t \in \mathbb{R}^{190}$ concatenates heterogeneous quantities of differing magnitude and meaning: joint positions, velocities, accelerations, and a binary visibility flag (Section 4.3). Before any temporal modelling, the sequence is mapped by

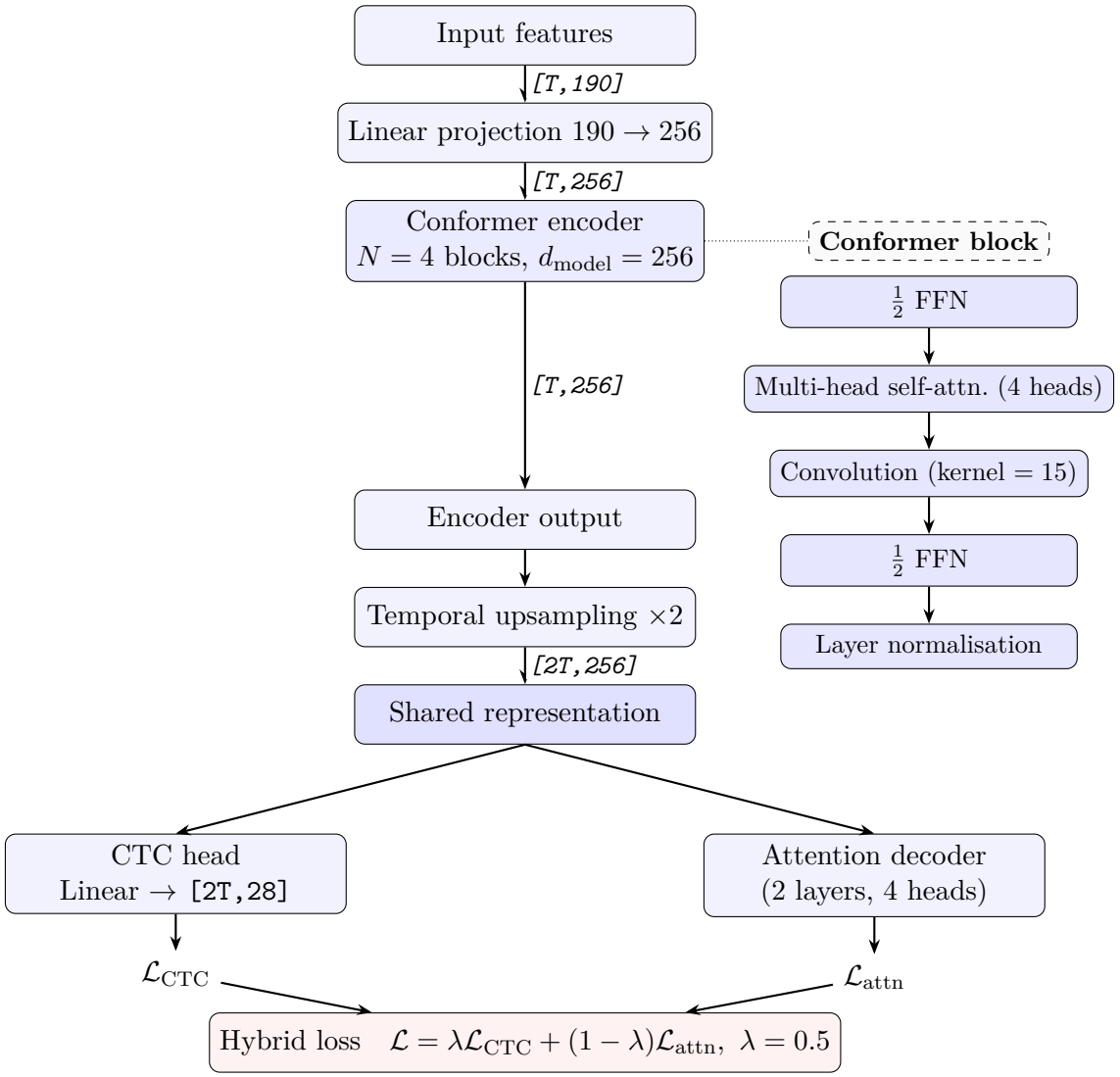

Figure 4.4: Detailed architecture of the Conformer recogniser with the joint CTC/attention head. The 190-dimensional input features are linearly projected to $d_{\text{model}} = 256$ and processed by $N = 4$ Conformer blocks; each block (right) applies a half-step feed-forward module, multi-head self-attention (4 heads), a convolution module (depthwise kernel size 15), a second half-step feed-forward module, and layer normalisation, each with a residual connection. The encoder output is upsampled in time by a factor of two to give the shared representation of length $2T$. This single representation feeds both the CTC head, which emits frame-level distributions over the 28-symbol vocabulary, and the autoregressive attention decoder; the branches are combined in the hybrid objective with $\lambda = 0.5$. Tensor shapes are shown at each stage.

a single learned linear layer into a homogeneous latent space of dimension $d_{\text{model}} = 256$,

$$\mathbf{H}^{(0)} = \tilde{\mathbf{F}}\,\mathbf{W}_{\text{in}} + \mathbf{b}_{\text{in}}, \qquad \mathbf{W}_{\text{in}} \in \mathbb{R}^{190\times 256},$$

so the input `[T, 190]` becomes `[T, 256]`. The projection places the heterogeneous feature channels on a common, learnable footing and lifts the low-dimensional tra-

jectory into a width suited to multi-head attention and the convolution module; $d_{\text{model}} = 256$ gives 64 dimensions per attention head while remaining modest enough to limit memorisation of writer-specific detail under the person-disjoint protocol. Relative positional information is supplied to the self-attention modules, so attention depends on inter-frame distance rather than absolute index, which suits words of varying length.

4.6.2 Conformer encoder

A written word carries information at two scales at once. Within a letter, the discriminative content is local: the curvature and velocity of a short segment of the trajectory. Across the word, the content is global: the order in which letters are produced. A model for air-writing must therefore capture both fine local stroke dynamics and long-range structure. The Conformer block is adopted because it combines the two in a single unit, with multi-head self-attention modelling global, long-range dependencies and a depthwise convolution modelling local temporal context.

The encoder stacks $N = 4$ Conformer blocks of width $d_{\text{model}} = 256$. Each block places a multi-head self-attention module and a convolution module between a macaron pair of half-step feed-forward modules,

$$\begin{aligned} \tilde{h} &= h + \tfrac{1}{2}\,\mathrm{FFN}(h), & h' &= \tilde{h} + \mathrm{MHSA}(\tilde{h}), \\ h'' &= h' + \mathrm{Conv}(h'), & h_{\text{out}} &= \mathrm{LayerNorm}\Big(h'' + \tfrac{1}{2}\,\mathrm{FFN}(h'')\Big). \end{aligned} \tag{4.4}$$

The two half-step feed-forward modules expand each frame representation and project it back, adding position-wise capacity at the start and end of the block. The multi-head self-attention module, with 4 heads, relates each frame to every other frame, allowing the encoder to reason about letter ordering across the whole word regardless of its length. The convolution module is the standard Conformer module; its depthwise kernel of size 15 gives each output frame a local receptive field spanning roughly a single stroke, capturing the local dynamics that self-attention alone does not emphasise. Residual connections and layer normalisation wrap every module, and dropout of 0.2 regularises the encoder for writer-independent generalisation. Each block maps `[T, 256]` to `[T, 256]`, enriching every per-frame representation with both local stroke and global word context, and after four blocks the output is a contextualised sequence $\mathbf{H} \in \mathbb{R}^{T \times 256}$. Four blocks suffice for the modest sequence length and low-dimensional input, avoiding the over-parameterisation that would heighten overfitting to the training writers.

4.6.3 Temporal upsampling

The encoder output is upsampled along the time axis by a factor of two, mapping `[T, 256]` to `[2T, 256]` with $T_{\text{out}} = 2T = 128$. This step provides alignment capacity for the CTC head. Under the collapse rule (Chapter 2), a repeated character can be produced only if a blank separates the two occurrences, so a target of length U with repeats needs at least $2U - 1$ frame positions; when the frame count is close to the target length, the monotonic alignment is tightly constrained and longer words may admit no valid alignment. Upsampling the encoder output doubles the alignment positions relative to the encoded length, providing room for blanks and repeats across the full range of word lengths. Its effect is quantified by ablation in §5.4.

4.6.4 CTC recognition head

The CTC branch is a linear layer that maps each of the T_{out} encoder states to a distribution over an output vocabulary of 28 symbols: the 26 lower-case letters, a blank symbol, and a repeat separator. The branch therefore produces frame-level logits of shape `[2T, 28]`, a categorical distribution over the vocabulary at each of the 128 positions. The blank symbol allows the model to emit "no character" at a frame and, as described above, separates repeated characters under the collapse rule. At training time the branch is optimised with the CTC loss [2] introduced in Chapter 2, which marginalises over all frame-to-label alignments consistent with the target; at inference the frame distributions are reduced to a string by greedy collapsing, merging repeats and removing blanks, as detailed in Section 4.8. CTC is well suited to air-writing because the continuous trajectory provides no frame-to-character segmentation and no pen-up signal to mark character boundaries; the loss learns the alignment implicitly and requires no manual segmentation of the input.

4.6.5 Attention decoder

In parallel with the CTC branch, an attention-based decoder transcribes the encoder output autoregressively, predicting each character conditioned on the previously emitted characters and on the encoder states,

$$p(\mathbf{y} \mid \tilde{\mathbf{F}}) = \prod_{u} p(y_u \mid y_{<u}, \mathbf{H}).$$

The decoder has 2 layers and 4 attention heads, with a feed-forward expansion factor of 4 and dropout 0.2. At each step, encoder–decoder attention lets the decoder

select the encoder states relevant to the next character, while self-attention over the previously generated characters models inter-character dependencies, that is, an implicit language model over the output alphabet. Decoding operates over the character set extended with start-of-sequence, end-of-sequence, and padding symbols: generation begins from the start-of-sequence symbol and continues until the decoder emits end-of-sequence. The branch is trained with a cross-entropy loss $\mathcal{L}_{\text{attn}}$. The attention decoder complements CTC in a specific way: whereas CTC treats the frame-level outputs as conditionally independent given the encoder, the attention decoder explicitly conditions each character on its predecessors, supplying the label-context modelling that CTC omits.

4.6.6 Joint CTC-attention training

The two branches are trained together because their inductive biases are complementary. CTC imposes a strictly monotonic, left-to-right alignment between frames and characters and provides dense frame-level supervision, but it assumes the outputs are conditionally independent given the encoder and so models little dependency between characters. The attention decoder models such dependencies directly, but its alignment is unconstrained and can be unstable to learn, particularly for longer sequences. Training both on a shared encoder combines their strengths: the monotonic alignment enforced by CTC regularises the encoder and provides a stable alignment signal, while the attention branch contributes character-level context. Because a single encoder must serve both an alignment-free frame classifier and a context-dependent decoder, it is driven towards representations that are robust rather than specialised to either objective. This multi-task regularisation is valuable under the person-disjoint protocol, where representations tuned too closely to the training writers would not transfer to unseen ones. An ablation removing either head (§5.4) confirms this complementarity.

The two losses are combined as

$$\mathcal{L} = \lambda \, \mathcal{L}_{\text{CTC}} + (1 - \lambda) \, \mathcal{L}_{\text{attn}},$$

the hybrid objective of Equation 2.4. The weight λ balances the two contributions: a larger λ emphasises the monotonic alignment and conditional-independence regularisation of CTC, while a smaller λ emphasises the label-context modelling of the attention decoder. The value $\lambda = 0.5$ gives the two objectives equal influence, so that the shared encoder is shaped equally by the alignment signal and the context model. Label smoothing of 0.1 is applied to the attention cross-entropy, discouraging over-confident character predictions and supporting writer-independent generalisation.

4.7 Training procedure

The model is trained for 80 epochs with the AdamW optimiser, using a peak learning rate of 5×10^{-4} under a one-cycle schedule, weight decay of 5×10^{-2}, gradient clipping at a norm of 1.0, and a batch size of 32, on a single GPU. A fixed random seed is used for reproducibility. Both output branches are optimised simultaneously under the joint objective of Section 4.6, so each gradient step updates the shared encoder through both losses.

Label-preserving trajectory augmentation is applied to the training split only. It comprises temporal warping (a global time-stretch of up to $\pm 15\%$), mild spatial jitter on the landmark coordinates (Gaussian noise of standard deviation 0.02 in normalised units), and temporal crop-and-resize (selecting a contiguous sub-segment of 60 to 100% of the clip and interpolating it back to T frames). Horizontal mirroring is deliberately excluded, because flipping the trajectory reverses the writing direction and corrupts the label. The validation and test splits are never augmented.

4.8 Decoding strategy

The trained model provides two decoders. The CTC branch is decoded greedily by taking the most probable symbol at each timestep, collapsing repeated symbols, and removing blanks. The attention branch is decoded greedily and autoregressively until the end-of-sequence symbol is emitted.

Rather than combining the two, the system selects a single decoder: the decoder with the lower character error rate on the *validation* split is chosen, and that decoder alone is then applied once to the test split. Selecting on validation keeps the test evaluation a single measurement of a fully specified system. The comparison between the two decoders is presented in Chapter 5.

Chapter 5

Experiments and Results

This chapter evaluates the final landmark-based recognition system on the WiTA English benchmark. It reports the landmark progression that leads to the final configuration (§5.2), the performance of the final landmark–Conformer system on validation and test (§5.3), a comparison with prior WiTA systems (§5.5), an analysis of the recognition behaviour (§5.6), an evaluation of the decoding strategy (§5.7), and a note on computational cost (§5.8).

5.1 Experimental setup

All experiments use the WiTA English dataset under the person-disjoint $8:1:1$ split described in Chapter 3, and report character error rate (CER, Equation 3.1) separately for the lexical and non-lexical subsets and overall. The landmark recogniser is small (a few million parameters) and trains in approximately one hour on a single NVIDIA T4 GPU; MediaPipe landmark extraction runs on CPU and is cached once. Model selection (checkpoint and decode head) is performed on the validation split, and each reported system is evaluated exactly once on the test split. Before adopting the landmark representation, two alternative families (global foundation-model embeddings and static trajectory images) were evaluated and found unsuitable for fine fingertip motion and were not pursued further.

5.2 Landmark-based recognition results

To isolate the contributions of per-clip normalisation and temporal resolution, a 2×2 factorial crosses the feature representation (absolute coordinates versus per-clip normalised) with the temporal length (T=32 versus T=64). All four configurations are

Table 5.1: Factorial isolation of per-clip normalisation (N) and temporal resolution (T). Validation CER (overall), under an identical training protocol; the test split is reserved for the final system. Margins give the main-effect means.

Representation	T=32	T=64	row mean
Absolute coordinates	0.440	**0.216**	0.328
Per-clip normalised	0.442	0.230	0.336
column mean	0.441	0.223	

Main effect of temporal resolution: -0.218. Main effect of normalisation: $+0.008$. Interaction $N \times T$: $+0.012$.

trained under an identical protocol and a fixed seed, and validation CER is reported; the test split is reserved for the final system. Table 5.1 reports the results.

The factorial is unambiguous. The *temporal resolution* accounts for almost the entire improvement: increasing T from 32 to 64 reduces validation CER by 0.218 on average, and does so at both representations ($0.440 \rightarrow 0.216$ for absolute coordinates and $0.442 \rightarrow 0.230$ for normalised ones). *Per-clip normalisation*, by contrast, has no measurable effect: its main effect is +0.008, within the run-to-run variation observed between training schedules, and at T=64 the absolute-coordinate model (0.216) is in fact marginally better than the normalised one (0.230). The small interaction (+0.012) does not alter this picture. The decisive factor is therefore the *temporal resolution* of the fingertip trajectory, not its normalisation.

5.2.1 Baseline landmark model

The Conformer CTC/attention model trained on absolute landmark coordinates at T=32 reaches a validation CER of approximately 0.44, and normalising the coordinates leaves it there (0.442). At this temporal resolution the landmark representation is learnable but limited: the short temporal window under-samples the trajectory, particularly for longer words.

5.2.2 Effect of per-clip normalisation

Per-clip normalisation (Equation 4.3) was introduced to remove a writer's absolute position and scale, in the expectation that it would improve writer-independent recognition. The factorial does not support this expectation: at both temporal resolutions the normalised and absolute models reach essentially the same validation CER (Table 5.1), and the main effect of normalisation (+0.008) is statistically indistinguishable from zero. The step is retained in the final system as a harmless, standard pre-

processing operation, but it is not responsible for the system's accuracy. Its (absence of) effect on the per-signer distribution is examined in §5.6.3.

5.2.3 Effect of temporal resolution (T=32 vs T=64)

Re-extracting the landmark cache at twice the temporal resolution (T=64, Section 4.5) is the change that moves the model off the plateau, independently of normalisation. Validation CER falls from ≈ 0.44 to ≈ 0.22, and the final (T=64) system reaches a test CER of 0.219 (§5.3.3). The improvement is consistent with the role of the extra temporal resolution in resolving longer words (§5.6.2) and with the dataset statistics of Chapter 3: English clips contain on average 78.75 native frames, more than twice the T=32 sampling that the plateau configurations used.

5.3 Final landmark–Conformer system

5.3.1 Final configuration

The final system combines per-clip normalised landmark trajectories at T=64 with the Conformer encoder and joint CTC/attention head of Chapter 4. The decode head is selected on the validation split and applied once to the test split (§5.7).

5.3.2 Validation results

Figure 5.1 shows the validation CER over training. The curve descends below the published baseline and stabilises, with the best checkpoint at epoch 59 (validation CER 0.226). At that checkpoint the attention head attains a validation CER of 0.266 and the CTC head 0.278; the attention head is therefore selected for test-time decoding.

5.3.3 Test results

Evaluated once on the test split with the validation-selected attention head, the final system attains an overall CER of 0.219, with 0.205 on lexical words and 0.310 on non-lexical strings. The per-head test results are reported in Table 5.6 and analysed in §5.7.

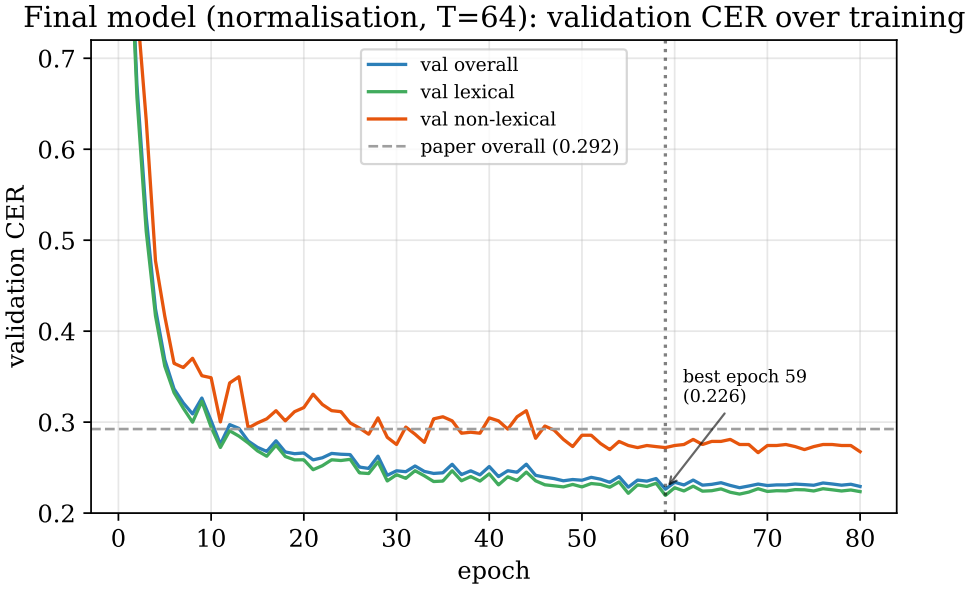

Figure 5.1: Validation CER over training for the final model (per-clip normalisation at T=64; overall, lexical, non-lexical). The best checkpoint is at epoch 59.

Table 5.2: Ablation of the recognition model (validation CER; lower is better). Each row changes one factor relative to the final configuration; all variants use normalised T=64 features and an identical training protocol.

Configuration	Change from final	Val CER
CTC-only (λ=1)	attention head removed	0.2754
Attention-only (λ=0)	CTC head removed	0.2589
Small encoder (d=128, L=2)	lower capacity	0.2490
No temporal upsampling ($\times$1)	encoder output not upsampled	0.2366
Final (d=256, L=4, λ=0.5, $\times$2)	—	**0.2301**
Large encoder (L=6)	higher capacity	0.2240

5.4 Ablation study

To justify the design of the recognition model (Chapter 4), each principal component is removed or altered while the rest of the system is held fixed. All variants use the same normalised T=64 features, an identical optimiser and schedule, and a fixed seed, and are trained for 60 epochs; validation CER is reported, since the test split is reserved for the final system. Table 5.2 summarises the results.

Joint CTC/attention objective. Removing either head degrades performance substantially: the attention-only model reaches 0.2589 and the CTC-only model 0.2754, both well above the joint model's 0.2301. The two objectives are therefore complementary rather than redundant, which justifies the joint training of §4.6: CTC contributes

a monotonic alignment and the attention head contributes label context, and neither alone matches their combination.

Temporal upsampling. Disabling the $\times 2$ temporal upsampling of the encoder output raises validation CER from 0.2301 to 0.2366. The additional alignment positions therefore help, consistent with the role of upsampling in giving CTC enough timesteps to place every character (§4.6).

Encoder capacity. Halving the width and depth (d=128, L=2) raises CER to 0.2490, whereas increasing the depth to L=6 lowers it only marginally to 0.2240. The chosen configuration (d=256, L=4) is thus a deliberate accuracy/complexity trade-off: most of the benefit of capacity is realised by four blocks, and the further gain from six blocks (0.006 CER) does not justify the additional parameters.

Scope. These ablations vary the recognition architecture at the final temporal resolution. The input factors — per-clip normalisation and temporal resolution — are isolated separately by the factorial of §5.2, which attributes the improvement to the temporal resolution.

5.5 Comparison with previous WiTA systems

Table 5.3 compares the proposed system with the two published WiTA English systems, the ST-R3D baseline of Kim et al. [1] and the Transformer-based TR-AWR of Tan et al. [7], all evaluated on the same person-disjoint test data under the same lexical/non-lexical CER metric. The proposed landmark–Conformer system reaches an overall test CER of 21.89%, a relative reduction of 25.1% over Kim et al. (29.24%) and 26.7% over Tan et al. (29.86%), achieved with an explicit low-dimensional landmark representation rather than full video and with the attention-based decoding that Kim et al. identify as future work. As the precise signer partition may differ between studies, the comparison is on the same dataset, protocol, and metric rather than on an identical split.

Table 5.3: Comparison with previous WiTA English systems (test-set CER, %; lower is better). Values for prior work are as reported in the respective papers.

Method	Representation	Architecture	Lexical	Non-lexical	Overall
Kim et al. (2023) [1]	RGB video	ST-R3D (3D-CNN) + CTC	28.10	36.46	29.24
Tan et al. (2023) [7]	RGB video	ViT + Transformer + CTC	28.65	37.51	29.86
Proposed	Landmark traj.	Conformer + CTC/attn	**20.46**	**30.98**	**21.89**

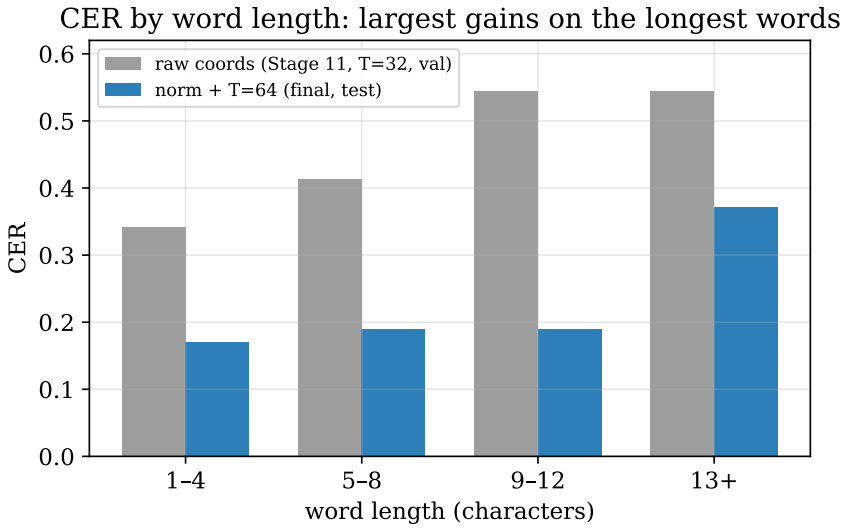

Figure 5.2: CER by word length, raw coordinates (T=32, validation) versus the final model (T=64, test). The largest gains are on long words.

5.6 Analysis of recognition performance

5.6.1 Lexical versus non-lexical recognition

Across all systems the non-lexical subset is harder than the lexical subset, because random strings cannot be supported by a language prior. The proposed system attains 0.205 on lexical and 0.310 on non-lexical words; the gap between the two is examined further through the decode heads in §5.7, where the attention head is found to favour lexical words and the CTC head to favour non-lexical strings.

5.6.2 Word-length analysis

Figure 5.2 reports CER by word length. The largest gains relative to the raw-coordinate model fall on longer words: words of 9–12 characters improve from 0.545 to 0.189. The longest words ($\geq$ 13 characters) remain the hardest (0.372) but are rare in the data. This pattern is consistent with the role of the higher temporal resolution in representing longer trajectories.

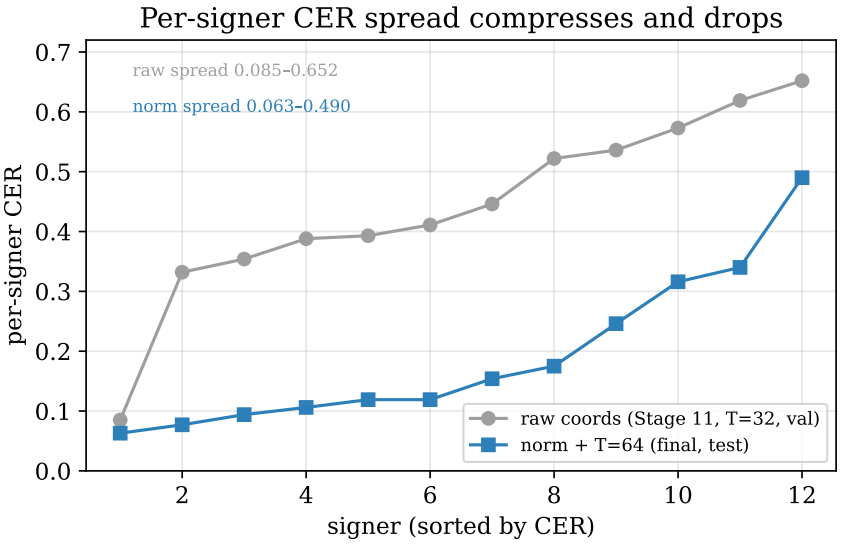

Figure 5.3: Per-signer CER, sorted. The final model lowers and tightens the distribution. The two models are evaluated on different person-disjoint subsets, so this compares distributions rather than paired signers.

5.6.3 Per-signer analysis

Figure 5.3 shows the per-signer CER distribution. Both the level and the spread improve: the raw-coordinate model ranges 0.085–0.652 across signers, whereas the final model ranges 0.063–0.490 with a lower median. Because this comparison also changes the temporal resolution, and the factorial of §5.2 attributes the overall gain to that factor, the tighter per-signer distribution is best read as a consequence of the higher temporal resolution rather than of normalisation. The hardest signer (`PSG`, 0.49) is also among the worst-tracked, linking the residual error to tracking quality rather than to the recogniser.

5.6.4 Qualitative error analysis

The aggregate metrics are complemented by a qualitative analysis of the 1,080 test transcriptions produced by the attention head. Of these, 42% are exactly correct; the remaining errors are decomposed by Levenshtein alignment into substitutions, insertions, and deletions in Table 5.4.

Substitutions and character confusions. Substitutions are the largest error class (70%) and they are highly structured. The most frequent confusions are between letters whose air-written trajectories are similar: $m \leftrightarrow n$, $u \leftrightarrow v \leftrightarrow w$, and the round letters o, e, c, and a (Figure 5.4). With no pen-up signal, such visually similar shapes are

Table 5.4: Decomposition of recognition errors by edit operation (attention head), as a percentage of all edits, with the corresponding CER.

Subset	Substitutions	Insertions	Deletions	CER
Overall	69.8%	19.2%	11.1%	0.219
Lexical	69.4%	18.9%	11.7%	0.205
Non-lexical	71.4%	20.3%	8.3%	0.310

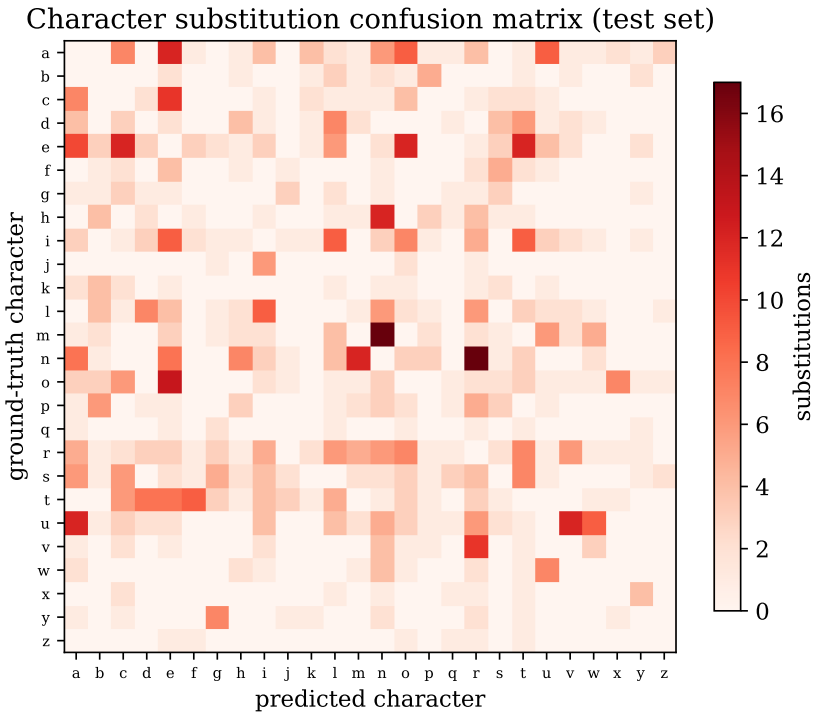

Figure 5.4: Character substitution confusion matrix on the test set (ground-truth row versus predicted column; the diagonal is empty by construction). The brightest off-diagonal cells correspond to kinematically similar letters such as m/n, $u/v/w$, and the round letters.

genuinely hard to separate from the trajectory alone, which is the dominant residual error of the system.

Insertions and the repeat separator. Insertions account for 19% of errors, and a large share of these are a decoding artifact rather than a genuine mistake. The recogniser uses a repeat-separator symbol to represent doubled letters; in 14% of predictions this symbol survives into the output string, where it is scored as a spurious insertion. Removing it in post-processing — the treatment the CTC blank already receives — would reduce the test CER from 0.219 to 0.198 without retraining. This simple correction is identified as immediate future work.

Table 5.5: Representative test predictions (attention head).

Ground truth	Prediction	Edit operations	Category
`roberts`	`roberts`	—	correct
`cumulative`	`amulative`	1 sub, 1 del	confusion + deletion
`bkwv`	`pkwv`	1 sub (b/p)	non-lexical confusion
`oexnn`	`oexn`	1 deletion	non-lexical deletion

Table 5.6: Test CER by decode head for the final model. The attention head is selected on validation and used for the headline result.

Decode	Overall	Lexical	Non-lexical
CTC-only	0.2328	0.2242	0.2874
Attention (selected)	**0.2189**	**0.2046**	0.3098

Length. The fraction of clips containing any error rises monotonically with word length, from 43% for words of 1–4 characters to 82% for words of 13 or more (cf. §5.6.2), as errors compound over the longer, more self-overlapping trajectories.

Non-lexical strings. The attention head might be expected to "correct" random strings towards real words, but this is rare: only 0.6% of non-lexical predictions are dictionary words. The non-lexical disadvantage of the attention head (§5.7) is therefore a character-level effect rather than whole-word hallucination. Representative cases are shown in Table 5.5.

5.7 Decoding strategy evaluation

Table 5.6 reports the test CER of each decode head. An initial evaluation selected, per clip, whichever head was closer to the ground truth; because this uses the reference, it is an oracle that is not realisable at test time and overstates performance. The reported protocol instead selects a single head on the validation split (§5.3.2) and applies it once to the test split. The two heads are complementary (Figure 5.5): the attention head, with its implicit language model, is stronger on lexical words (0.205 versus 0.224), while the CTC head is stronger on non-lexical strings (0.287 versus 0.310). The validation-selected attention head is used for all headline results; notably, the non-selected CTC head (0.233 overall) also improves on the published baseline, so the result does not depend on the head selection.

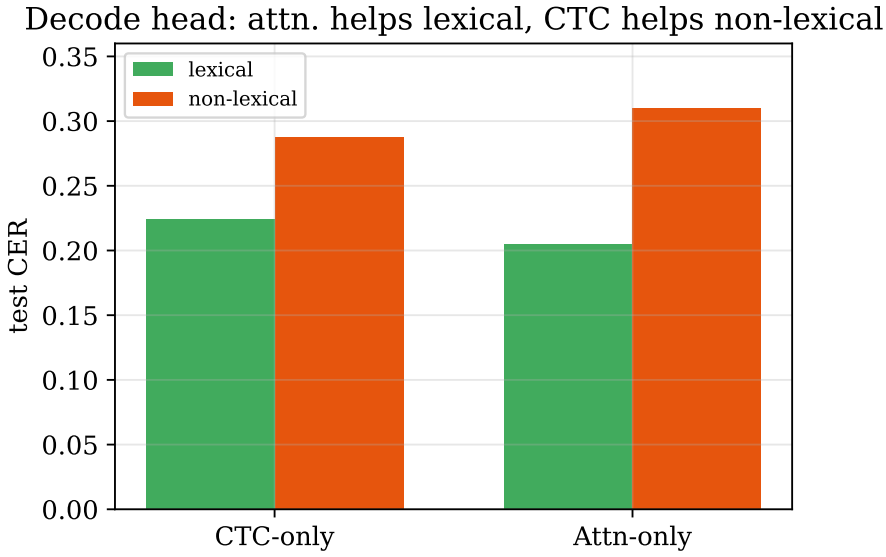

Figure 5.5: Decode head versus subset. The attention head favours lexical words and the CTC head favours non-lexical strings.

5.8 Computational performance

For reference, Kim et al. report a decoding throughput of 697.39 frames per second for ST-R3D and Tan et al. report 194.67 D-fps for TR-AWR on their respective hardware. The proposed recogniser is comparatively small (a few million parameters) and operates on a 190-dimensional landmark sequence rather than on video, so the dominant runtime cost is the MediaPipe landmark extraction, which runs in real time. A directly comparable decoding-throughput measurement on matched hardware was not conducted and is identified as future work.

5.9 Summary of findings

These experiments converge on a single conclusion: on WiTA, *representation, not architecture, governs accuracy.* The decisive input change, the increase in temporal resolution to T=64, moves the CER from ≈ 0.45 to 0.22, whereas every architectural ablation moves it by at most 0.045 (§5.4); and the resulting trajectory system outperforms both published RGB-video baselines. The individual findings that support this conclusion are:

- Landmark trajectories are sufficient for accurate unconstrained air-writing recognition: the final landmark system attains an overall test CER of 0.219 without using raw video.

- Increasing the temporal resolution from T=32 to T=64 is the change that moves the landmark model off the ≈ 0.45 plateau, with the largest gains on longer words.
- A controlled factorial isolating per-clip normalisation and temporal resolution attributes the entire improvement to the temporal resolution; per-clip normalisation has no measurable effect on CER.
- The final landmark–Conformer system is the best-performing system in this dissertation and improves on both published WiTA English systems, by 25.1% relative overall against Kim et al. [1] and 26.7% against Tan et al. [7].
- The CTC and attention heads are complementary, and the result is robust to the choice of head: even the non-selected head improves on the published baseline.

Chapter 6

Conclusion and Future Work

6.1 Summary

This dissertation studied air-writing recognition on the English WiTA benchmark, with the goal of improving on the published baseline character error rate (CER) of 0.292. We proposed two changes that act in combination. The first is a better input *representation*: an explicit fingertip-trajectory sequence, comprising position, velocity, and acceleration extracted from hand landmarks at a temporal resolution of T=64, in place of raw RGB video. The second is a compact recognition *architecture*: a Conformer encoder with a joint CTC/attention head, which adds the attention-based decoding that Kim et al. left as future work. Together these attain a test CER of **0.219**, beating the published video baselines of Kim et al. (0.292) and Tan et al. (0.299) by 15–27% relative, while using a model of only a few million parameters operating on a 190-dimensional landmark sequence rather than full video, and a fair decode selected on validation. A controlled factorial confirms that the temporal resolution of the trajectory is the property responsible for most of this gain, with per-clip normalisation contributing no measurable effect.

6.2 Limitations

Three caveats qualify these results. (i) **Evaluation:** during development, the discipline of fixing all decoder choices on validation surfaced an oracle bug in an initial per-clip head selection; the reported headline uses the corrected, deployable decode, and all heads are reported. (ii) **Split:** the comparison to the published systems is on the same WiTA person-disjoint protocol and metric, but the precise signer partition may differ between studies, so the comparison is on dataset, protocol, and metric

rather than on an identical split. (iii) **Stochasticity:** with twelve signers per split, per-split CER carries non-trivial variance; we mitigate over-interpretation by checking that even the non-selected decode head beats the baseline and by reporting per-signer and per-length breakdowns rather than a single number.

6.3 Future work

Several directions follow naturally.

- **Hybrid representations.** While landmark trajectories provide the strongest signal in the current study, future work could investigate hybrid architectures that combine trajectory features with appearance-based hand representations. Such models may retain the temporal advantages of landmarks while recovering visual cues lost during landmark extraction.
- **Adaptive temporal sampling.** The factorial analysis identifies temporal resolution as the most influential design factor. Instead of using a fixed number of frames, future work could investigate adaptive sampling strategies that allocate more temporal detail to longer or more complex writing motions.
- **Language-model integration.** The current decoding process relies primarily on the recognition model outputs. Future work could incorporate character-level or word-level language models to improve decoding robustness and reduce errors in ambiguous character sequences.
- **Continuous sentence-level air-writing.** The WiTA benchmark focuses on isolated words. Future work could extend the proposed framework to continuous air-writing recognition, where word boundaries are unknown and recognition must be performed over longer unconstrained sequences.

Bibliography

[1] G. Kim et al., "WiTA: Writing in The Air — a benchmark and baseline for air-writing recognition," *IEEE Trans. Artif. Intell.*, 2023.

[2] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. Int. Conf. Machine Learning (ICML)*, 2006, pp. 369–376.

[3] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2015.

[4] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 4835–4839.

[5] A. Gulati et al., "Conformer: Convolution-augmented Transformer for speech recognition," in *Proc. Interspeech*, 2020, pp. 5036–5040.

[6] F. Zhang et al., "MediaPipe Hands: On-device real-time hand tracking," *arXiv preprint arXiv:2006.10214*, 2020.

[7] X. Tan, J. Tong, T. Matsumaru, V. Dutta, and X. He, "An end-to-end air writing recognition method based on Transformer," *IEEE Access*, vol. 11, pp. 109885–109898, 2023.

[8] C. Amma, M. Georgi, and T. Schultz, "Airwriting: A wearable handwriting recognition system," *Personal Ubiquitous Comput.*, vol. 18, no. 1, pp. 191–203, 2014.

[9] M. Sepahvand, F. Abdali-Mohammadi, and F. Mardukhi, "Evolutionary metric-learning-based recognition algorithm for online isolated Persian/Arabic characters, reconstructed using inertial pen signals," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2872–2884, 2017.

[10] X. Zhang, Z. Ye, L. Jin, Z. Feng, and S. Xu, "A new writing experience: Finger writing in the air using a Kinect sensor," *IEEE MultiMedia*, vol. 20, no. 4, pp. 85–93, 2013.

[11] S. Mohammadi and R. Maleki, "Real-time Kinect-based air-writing system with a novel analytical classifier," *Int. J. Document Anal. Recognit.*, vol. 22, no. 2, pp. 113–125, 2019.

[12] P. Kumar, R. Saini, P. P. Roy, and D. P. Dogra, "Study of text segmentation and recognition using Leap Motion sensor," *IEEE Sensors J.*, vol. 17, no. 5, pp. 1293–1301, 2017.

[13] M. Arsalan, A. Santra, K. Bierzynski, and V. Issakov, "Air-writing with sparse network of radars using spatio-temporal learning," in *Proc. Int. Conf. Pattern Recognition (ICPR)*, 2021, pp. 8877–8884.

[14] Z. Fu, J. Xu, Z. Zhu, A. X. Liu, and X. Sun, "Writing in the air with WiFi signals for virtual reality devices," *IEEE Trans. Mobile Comput.*, vol. 18, no. 2, pp. 473–484, 2019.

[15] M. Chen, G. AlRegib, and B.-H. Juang, "Air-writing recognition—Part I: Modeling and recognition of characters, words, and connecting motions," *IEEE Trans. Human-Mach. Syst.*, vol. 46, no. 3, pp. 403–413, 2016.

[16] A. Schick, D. Morlock, C. Amma, T. Schultz, and R. Stiefelhagen, "Vision-based handwriting recognition for unrestricted text input in mid-air," in *Proc. ACM Int. Conf. Multimodal Interaction*, 2012, pp. 217–220.

[17] S. Mukherjee, S. A. Ahmed, D. P. Dogra, S. Kar, and P. P. Roy, "Fingertip detection and tracking for recognition of air-writing in videos," *Expert Syst. Appl.*, vol. 136, pp. 217–229, 2019.

[18] H. J. Chang, G. Garcia-Hernando, D. Tang, and T.-K. Kim, "Spatio-temporal Hough forest for efficient detection-localisation-recognition of fingerwriting in egocentric camera," *Comput. Vis. Image Underst.*, vol. 148, pp. 87–96, 2016.

[19] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 855–868, 2009.

[20] X.-Y. Zhang, Y.-M. Zhang, C.-L. Liu, and Y. Bengio, "Drawing and recognizing Chinese characters with recurrent neural network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 849–862, 2018.

[21] G. Bastas, K. Kritsis, and V. Katsouros, "Air-writing recognition using deep convolutional and recurrent neural network architectures," in *Proc. Int. Conf. Frontiers in Handwriting Recognition (ICFHR)*, 2020, pp. 7–12.

[22] M. M. Alam, M. T. Islam, and S. M. Rahman, "Trajectory-based air-writing recognition using deep neural network and depth sensor," *Sensors*, vol. 20, no. 2, art. 376, 2020.

[23] J. Gan, W. Wang, and K. Lu, "A unified CNN-RNN approach for in-air handwritten English word recognition," in *Proc. IEEE Int. Conf. Multimedia and Expo (ICME)*, 2018, pp. 1–6.

[24] J. Gan and W. Wang, "In-air handwritten English word recognition using attention recurrent translator," *Neural Comput. Appl.*, vol. 31, pp. 3155–3172, 2019.

[25] T. Ahmad, L. Jin, X. Zhang, S. Lai, G. Tang, and L. Lin, "Graph convolutional neural network for human action recognition: A comprehensive survey," *IEEE Trans. Artif. Intell.*, vol. 2, no. 2, pp. 128–145, 2021.

Appendix A

Hyperparameters and Reproducibility

Table A.1: Hyperparameters of the final landmark model.

Component	Value
Input feature dimension	190 (pos $\oplus$ vel $\oplus$ acc $\oplus$ vis)
Native / resampled length T	64 (frame budget 128)
Encoder	Conformer, 4 blocks
Model width d_{model}	256
Attention heads	4
Convolution kernel	15
Temporal upsample	$\times 2$
CTC vocabulary	28 (26 letters, blank, separator)
Attention decoder	2 layers, 4 heads
Joint loss weight λ	0.5
Label smoothing	0.1
Optimiser	AdamW
Peak learning rate	5×10^{-4} (one-cycle)
Weight decay	5×10^{-2}
Gradient clip	1.0
Batch size	32
Epochs	80 (best at 59)
Augmentation (train only)	temporal warp, spatial jitter, temporal crop

Normalisation. Each clip is centred and scaled (Equation 4.3) using only that clip's statistics; the same scale is applied to position, velocity, and acceleration channels. **Selection.** The checkpoint is chosen by validation overall CER (epoch 59); the decode head (attention) is chosen by validation CER; the test set is evaluated once.

Appendix B

Detailed Test-Set Results

Table B.1 gives the per-signer test CER of the final model for all twelve test signers, and Table B.2 the CER by word length. The hardest signer (`PSG`) is also among the worst-tracked (Figure 3.2), and the hardest length bucket (≥ 13 characters) is rare in the data.

Table B.1: Per-signer test CER of the final model (12 test signers, sorted).

Signer	CER	Signer	CER
CHJ	0.063	DK	0.154
SME	0.077	HJM	0.175
YJE	0.094	KJH	0.246
PMY	0.106	HSJ	0.316
PJH	0.119	JTH	0.340
KHJ	0.119	PSG	0.490
min 0.063	median ≈ 0.14	max 0.490	spread 0.427

Table B.2: Test CER by word-length bucket for the final model, with the share of clips in each bucket.

Word length (chars)	Test CER	relative difficulty
1–4	0.170	easiest
5–8	0.189	
9–12	0.189	(was 0.545 with raw coords)
13+	0.372	hardest, rare