Mitigating Exposure Bias in IMU Handwriting Recognition with Noise Injection and Hybrid CTC–AR Training

Muhammad Ajlal[1⋆[0009−0003−5966−6130]], Jindong Li[1,3[0000−0002−3550−1660]], Dario Zanca[1[0000−0001−5886−0597]], Vincent Christlein[1[0000−0003−0455−3799]], and Björn Eskofier[1,2,3,4[0000−0002−0417−0336]]

[1] Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany
muhammad.ajlal@fau.de
[2] Ludwig-Maximilians-Universität München, Munich, Germany
[3] Munich Center for Machine Learning, Munich, Germany
[4] Helmholtz Zentrum München – German Research Center for Environmental Health, Neuherberg, Germany

Abstract. Handwriting recognition (HWR) from inertial measurement unit (IMU) signals uses pen motion recorded directly by on-device sensors, offering a privacy-conscious alternative to camera-based or digitizing-tablet capture. Current IMU-based recognizers, however, remain dominated by recurrent models trained with connectionist temporal classification (CTC), while attention-based autoregressive (AR) recognizers are still largely unexplored for this modality. We introduce HWRFormer, the first AR-based model for IMU-based HWR, as an encoder–decoder recognizer that combines a 1D convolutional neural network (CNN) encoder with an AR Transformer decoder. The migration to AR decoding improves word-level recognition, but teacher-forced training can weaken character-level robustness when predicted prefixes drift at inference time. We address this exposure-bias effect with two complementary interventions. Noise injection perturbs the teacher-forced label prefix during training, while hybrid CTC–AR training adds training-only auxiliary alignment supervision to the encoder. On the public OnHW-words500 writer-independent (WI) split, HWRFormer improves both character error rate (CER) and word error rate (WER) over the recurrent CTC baseline; on large-vocabulary private words and long-context private sentences, it substantially reduces WER, while CER remains behind the recurrent baseline, reflecting a trade-off rather than a uniform improvement. Noise injection improves decoder-prefix robustness, while hybrid CTC–AR training is most effective on OnHW word recognition. Code is available at https://github.com/muhammadajlal/HWRFormer.

Keywords: IMU-based HWR · Exposure bias · Hybrid CTC–AR training · Noise injection · Writer-independent recognition

⋆ Corresponding author.

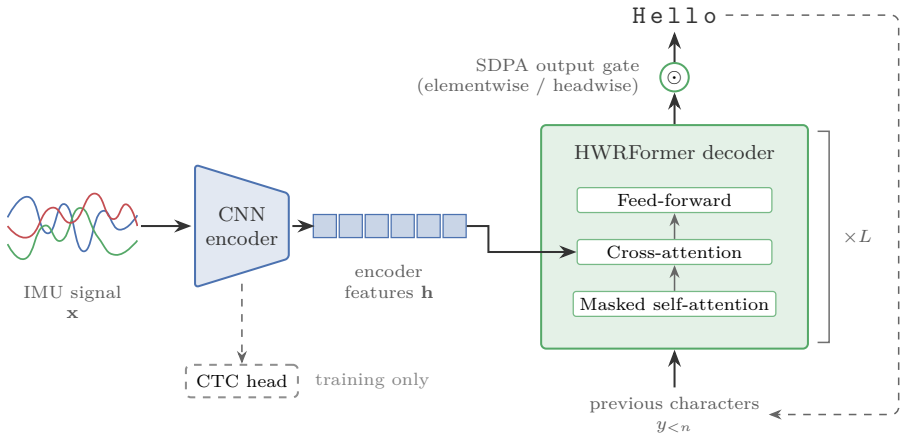

Fig. 1. HWRFormer architecture and hybrid training. At inference, the shared 1D CNN encoder feeds an AR Transformer decoder, and the decoder consumes its own previous predictions (dashed feedback loop). Scaled dot-product attention (SDPA) output gating is applied to the attention outputs within each decoder layer and is shown schematically at the decoder output. During hybrid training, an auxiliary CTC head on the same encoder (dashed, training only) adds an alignment loss summed with the AR loss.

1 Introduction

Pen-based document systems are re-emerging as a practical input modality, driven by smart pens with IMUs embedded in the barrel [16,11]. Unlike camera- or tablet-based systems, these pens record pen motion and contact force rather than page images. The raw sensor traces are not anonymous by themselves—writer identification has been demonstrated directly from inertial handwriting signals [7]—but on-device inference can avoid transmitting them, making the modality attractive for clinical screening [20], classroom-based dysgraphia assessment [5], and other privacy-sensitive settings.

The recognition task maps a multi-channel IMU time series to the underlying character string. The dominant IMU-based HWR pipeline still uses a 1D CNN encoder and a bidirectional long short-term memory (BiLSTM) recognizer trained with CTC loss [8,16,11]. This leaves open whether attention-based AR decoding, which is widely used in neighboring sequence-recognition tasks [21,6,14], can improve IMU-based HWR under a matched parameter budget; we therefore study a controlled migration from the recurrent CTC baseline to an attention-based AR recognizer.

AR Transformer decoders remove CTC's frame-independence assumption by conditioning each character on the generated prefix. This larger context can improve word-level recognition because ambiguous local evidence can be interpreted with neighboring characters. However, it also introduces exposure bias: the decoder receives the ground-truth previous token during training but its own

previous prediction during inference. When an early prediction is wrong, later characters are conditioned on a prefix that was not observed during training. Common remedies either perturb the decoder prefix, as in scheduled sampling or noise-injection methods, or add auxiliary alignment pressure to strengthen encoder evidence.

We propose **HWRFormer**, the first AR-based model for IMU-based HWR, as a complete encoder–decoder recognizer that combines the established 1D CNN encoder with an AR Transformer decoder and SDPA output gating (Fig. 1). In our experiments, the migration to AR decoding appears as a CER/WER trade-off: HWRFormer improves word-level recognition, but can weaken character-level robustness when predicted prefixes drift. We therefore study two complementary ways to make this migration robust. The paper makes three contributions.

1. A controlled migration from the CNN-BiLSTM baseline trained with CTC loss to an attention-based AR recognizer under a matched parameter budget. A CNN-Transformer-CTC baseline separates the effect of self-attention from that of AR decoding, and SDPA output gating is evaluated as a lightweight architectural variant.
2. Noise injection, a regularizer for AR training that perturbs the decoder's input tokens while keeping teacher forcing on the loss side. We evaluate five replacement modes and a rate sweep across four word and sentence settings.
3. Hybrid CTC–AR training [22,10] as a targeted auxiliary-alignment intervention. We test whether this training-only encoder supervision improves small-vocabulary word recognition.

2 Related Work

2.1 IMU-based HWR

Online HWR has relied on tablet- or stylus-based capture of pen-position traces [17]; IMU-equipped pens remove the need for a specialized writing surface.

Wehbi et al. [23] introduced the first end-to-end deep model for the STABILO DigiPen, pairing 1D convolutions with a BiLSTM under a CTC objective and achieving word-level recognition without a dictionary or language model. The OnHW benchmark released by Ott et al. [16] subsequently established the large-scale WI evaluation protocol now used in the field, including the OnHW-words500 dataset. The recent Robust and Efficient Writer-Independent (REWI) architecture [11] refines the same CNN-BiLSTM model family trained with CTC and reports state-of-the-art results on OnHW-words500, alongside evaluations on private datasets.

Alternative architectural choices have also been explored. Alemayoh et al. [1] combine inertial channels with tip-force sensors and compare CNN, LSTM, and Vision Transformer architectures for character-level recognition. These works advance sensor-based pen recognition, but the dominant high-performing systems for IMU-based HWR still rely on recurrent CTC recognition.

2.2 Autoregressive Transformers and Gated Attention

The Transformer [21] was introduced for machine translation and now underpins most large-scale sequence models. Encoder–decoder attention is also widely used in speech recognition [6,9] and offline handwriting [14,3], where autoregressive decoding can exploit textual context while attending to input evidence. This makes attention-based AR recognition a natural candidate for IMU-based HWR, even though the modality has mostly been studied with recurrent CTC models.

Gating mechanisms have been applied to Transformers at multiple positions. Gated activations in the feed-forward sublayer, such as the gated linear unit (GLU) variants of Shazeer [19], improve perplexity at no inference cost and are now common in many large language model (LLM) backbones. Qiu et al. [18] study gating at the SDPA output and show that a head-specific sigmoid gate improves perplexity, stabilizes training by suppressing loss spikes, and mitigates the attention-sink phenomenon. Their analysis links the gain to added non-linearity in the value-output projection and to input-dependent sparsity that filters redundant attention mass.

2.3 Exposure Bias in Autoregressive Sequence Models

Teacher-forced AR training is known to suffer from exposure bias. The decoder sees ground-truth prefixes during training, but must condition on its own predictions at inference time [2]. Existing remedies act either on the decoder input or on the encoder evidence.

On the decoder side, scheduled sampling [2] interpolates between teacher forcing and self-conditioning during training, but is unstable for Transformers [15]. Word dropout [4] and related noise-injection techniques perturb the decoder input directly while keeping teacher forcing on the loss side.

On the encoder side, joint CTC–attention training was introduced for end-to-end speech recognition by Watanabe et al. [22] and concurrently by Kim et al. [10]. The auxiliary CTC loss attaches frame-level alignment supervision to a shared encoder, which can stabilize attention training and accelerate encoder convergence. The technique is now standard in speech recognition, including the Conformer architecture [9]. In this paper we test whether the same alignment pressure can improve IMU-based HWR by preserving encoder evidence for an AR decoder.

3 Method

3.1 HWRFormer

HWRFormer is a complete encoder–decoder recognizer. Its encoder is the 1D CNN encoder from REWI [11], which we keep unchanged across comparisons so that the experiments isolate the decoder and training objective. It maps a 13-channel IMU sequence $\mathbf{x} \in \mathbb{R}^{T\times 13}$ to a downsampled feature sequence $\mathbf{h} \in \mathbb{R}^{T'\times 512}$.

The decoder is a two-layer Transformer with four attention heads, a 256-dimensional hidden state, and a feed-forward width of 896, attached to the encoder output $\mathbf{h}$. We evaluate two lightweight SDPA output gates following prior gated-attention work [18]: an elementwise sigmoid gate over features and a headwise scalar gate per attention head. The hyperparameters are chosen so that the full encoder–decoder model is comparable in parameter count to the REWI baseline.

AR Training HWRFormer is trained with cross-entropy under teacher forcing on character-level targets. The decoder vocabulary $\mathcal{V}_{\text{ar}}$ extends the CTC character vocabulary $\mathcal{V}_{\text{ctc}}$, which includes the blank, with padding (PAD), beginning-of-sequence (BOS), and end-of-sequence (EOS) tokens.

3.2 Noise Injection

Noise injection is a decoder-prefix regularizer that targets exposure bias while keeping teacher forcing on the loss side. Let $\mathbf{y} = (y_1, \ldots, y_N)$ be the ground-truth character sequence and $\mathbf{y}_{\text{inp}} = (\text{BOS}, y_1, \ldots, y_{N-1})$ the teacher-forced decoder input. For each mode m, the replacement distribution $q_m(x)$ maps an input character x to a probability distribution over replacement characters. We construct the noisy input $\tilde{\mathbf{y}}_{\text{inp}}$ by sampling independently at each non-special position t,

$$\tilde{y}_{\text{inp},t} \;=\; \begin{cases} \tilde{x}_t, \;\; \tilde{x}_t \sim q_m(y_{\text{inp},t}) & \text{with probability } p_{\text{ni}}, \\ y_{\text{inp},t} & \text{otherwise,} \end{cases} \tag{1}$$

and train the decoder with cross-entropy on the unchanged target $\mathbf{y}$ conditioned on $\tilde{\mathbf{y}}_{\text{inp}}$ and the encoder features $\mathbf{h}$:

$$\mathcal{L}_{\text{ar}}^{\text{ni}} \;=\; -\sum_{t=1}^{N} \log P_\theta\big(y_t \,\big|\, \tilde{\mathbf{y}}_{\text{inp},\le t},\, \mathbf{h}\big)\,. \tag{2}$$

The decoder is therefore trained to predict the next ground-truth token even when its prefix is partially wrong, encouraging it to rely on encoder evidence rather than only on the previous-step input. The mechanism is the character-level analogue of word dropout [4]. Noise injection adds no parameters and no inference cost. We use $p_{\text{ni}} = 0.15$ unless stated otherwise.

The five choices of q_m form four families: agnostic uniform noise, corpus-based bigram substitutions, recognizer-specific self-confusion, and position-level adjacent swaps. This lets the ablation test whether robustness comes from generic noise or from structured errors.

Uniform noise is the agnostic mode. It draws $\tilde{x}$ from $\mathcal{V}_{\text{ar}}$ after removing PAD, BOS, EOS, and the CTC blank, independently of x. Because it ignores linguistic plausibility and recognizer-specific error patterns, a positive effect indicates that dense prefix perturbation alone can help.

Bigram-right and bigram-left are corpus-based modes computed from character bigrams in the training fold. Bigram-right replaces x by characters that empirically follow x, i.e., $P(y_{t+1} \mid y_t{=}x)$, while bigram-left replaces x by characters that empirically precede x, i.e., $P(y_{t-1} \mid y_t{=}x)$. The pair tests whether forward or backward corpus statistics better approximate inference-time prefix errors under left-to-right decoding.

Self-confusion is recognizer-specific. It builds a per-fold confusion matrix by greedily decoding the HWRFormer trained without noise injection on the training fold; replacements are then sampled from the empirical distribution of predicted characters for each ground-truth character x. Bigram and self-confusion lookup tables use additive smoothing with weight 1.0 for unseen contexts.

Adjacent-swap is position-level. It perturbs order rather than identity by exchanging the token at t with a neighboring prefix token, modeling transposition-style errors while leaving the multiset of prefix characters unchanged.

All five modes share the same training loop and are selected by a single `mode` flag, so the comparison controls for everything except q_m.

3.3 Hybrid CTC–AR Model

The hybrid model attaches an auxiliary CTC head to the encoder output $\mathbf{h}$ in parallel with HWRFormer's AR decoder. The total loss is

$$\mathcal{L} = \lambda_{\text{ar}}\mathcal{L}_{\text{ar}} + \lambda_{\text{ctc}}\mathcal{L}_{\text{ctc}}, \tag{3}$$

with $\lambda_{\text{ar}} = 1.0$ and $\lambda_{\text{ctc}} = 0.1$ throughout, the empirical optimum from the sweep in Sect. 4.4. The auxiliary CTC head is used only during training. Decoding uses HWRFormer's AR decoder.

4 Experiments

4.1 Experimental Setup

Datasets The primary evaluation dataset is the public OnHW-words500 word split [16], evaluated under both the WI and writer-dependent (WD) protocols. The dataset comprises 25,199 samples from 53 writers, partitioned into five cross-validation folds: WI folds are writer-disjoint (each fold holds out the samples of roughly one fifth of the writers, giving approximately 20,160 training and 5,040 validation samples per fold), whereas WD splits by words so that all writers appear in both training and validation. The samples are German and English words recorded with a STABILO DigiPen, using 13-channel IMU traces from two accelerometers, a gyroscope, a magnetometer, and a force sensor. Labels come from a 500-word vocabulary and a 60-entry character set, consisting of upper- and lower-case English and German letters, and the CTC blank symbol.

We use this dataset to test small-vocabulary word recognition under different writer-overlap protocols.

In addition, we report results on a private STABILO IMU pen dataset recorded with an updated sensor hardware generation [11], evaluated under the WI protocol with writer-disjoint 5-fold splits. The dataset has a German and English word subset with 24,163 samples from 520 writers, using the same 60-entry German/English character set as OnHW-words500, and a sentence subset with 20,639 samples from 456 writers. The sentence subset contains longer sequences from a large vocabulary and extends the same 60-entry character set with the digits 0–9, the symbols `! ’ ( ) + , - . / : ; = ?` ·, and the space character, yielding 85 entries.

The private dataset cannot be redistributed for licensing reasons, so the public OnHW-words500 experiments form the reproducible core of this study; the private subsets serve as cross-dataset confirmation under different recording conditions and as the only long-context evaluation.

Training We adopt REWI's preprocessing pipeline unchanged [11] and follow its training paradigm: all models are trained for 300 epochs with AdamW [13], base learning rate 10^{-3}, cosine annealing schedule [12], and 30 warmup epochs. The mini-batch size is 64 and gradients are clipped to a norm of 5.0. The AR cross-entropy loss uses label smoothing of 0.1. Only the best-CER checkpoint is retained for downstream inference. All experiments use the predefined 5-fold splits provided by [16,11]. To ensure a fair comparison, the REWI baseline is scaled to match HWRFormer's parameter budget. The CNN front end, optimizer, and schedule are otherwise identical across models.

Metrics We report 5-fold validation means of CER and WER, both Levenshtein edit rates normalized by reference characters and words, respectively. Checkpoints minimize validation CER; WER and decoded-output analyses use the same epoch. Hyperparameter selection and reporting use the same folds, so sweep comparisons are descriptive.

We also report model size and inference cost using trainable parameters and multiply-accumulate operations (MACs), computed as half the floating-point operation count from PyTorch's `FlopCounterMode`. The MACs in Table 2 use the OnHW-words500 word setting with 1,024 timesteps. For AR and hybrid models, the counter runs the full greedy decoding path with len_max=6, the rounded-up mean OnHW-words500 text label length.

4.2 From CTC to AR Decoding

Table 1 uses the WI and WD splits of public OnHW-words500 as the primary migration benchmark and adds the private subsets as transfer checks. To separate the contribution of self-attention from that of AR decoding, we additionally evaluate a parameter-matched CNN-Transformer-CTC baseline, which replaces REWI's BiLSTM with a Transformer encoder.

Table 1. Controlled migration from REWI to HWRFormer on OnHW-words500 (OnHW) and the private word and sentence subsets: 5-fold mean and across-fold standard deviation (STD) of CER and WER in %; lower is better. Best mean per dataset and metric in **bold**.

Method	Metric	OnHW (WI)		OnHW (WD)		Priv. (words)		Priv. (sent.)	
		CER	WER	CER	WER	CER	WER	CER	WER
REWI [11]	Mean	7.30	15.16	14.81	44.77	**9.39**	31.82	**6.55**	23.52
	STD	6.85	10.05	3.84	7.84	0.59	0.69	0.71	2.27
CNN-Transf.-CTC	Mean	7.16	15.33	**12.96**	41.88	9.53	33.81	7.41	29.13
	STD	6.90	10.43	3.44	7.65	0.63	0.82	0.82	2.94
HWRFormer (ungated)	Mean	7.10	**10.39**	16.47	32.07	10.59	21.39	10.37	16.73
	STD	7.15	8.08	5.73	9.21	0.77	1.23	1.08	1.57
HWRFormer (elementw.)	Mean	**6.94**	10.50	16.31	31.87	9.96	**19.04**	9.28	**14.88**
	STD	7.10	8.26	6.20	9.31	0.44	0.56	1.50	1.89
HWRFormer (headw.)	Mean	6.99	10.47	15.70	**31.08**	9.63	19.07	9.12	14.96
	STD	7.19	8.42	5.91	9.25	0.71	1.08	1.27	1.47

Table 2. Model size and inference cost: trainable parameters and multiply-accumulate operations (MACs) in millions, measured as defined in Sect. 4.1. Lowest MACs in **bold**.

Method	#Params (M)	MACs (M)
REWI [11]	4.64	**413**
CNN-Transformer-CTC	4.62	429
HWRFormer (ungated)	4.57	653
HWRFormer (elementwise)	4.64	669
HWRFormer (headwise)	4.57	667

Autoregressive decoding and gating On public OnHW WI, replacing REWI with HWRFormer trained by AR cross-entropy cuts WER from 15.16 to 10.39 (−31.5 %), while CER changes only from 7.30 to 7.10. The parameter-matched CNN-Transformer-CTC baseline keeps both metrics near REWI (CER 7.16, WER 15.33), so replacing recurrence with self-attention alone does not explain the word-level gain. SDPA gating then provides a small further CER reduction (6.94 elementwise, 6.99 headwise) with negligible parameter increase. The WD split shows the same word-level effect. Headwise gating gives the best WER, reducing REWI's 44.77 to 31.08, a 13.69 pp absolute and 30.6 % relative reduction. This gain comes with higher inference cost because greedy AR decoding runs one step per output character. At len_max=6, the full AR decode costs 58–62 % more MACs than REWI's single-pass CTC decoding (Table 2).

Private-dataset evaluation Private results in Table 1 compare the HWRFormer variants with the CTC baselines. HWRFormer (elementwise) substan-

tially reduces REWI's WER on both subsets (31.82 to 19.04 on words, −40.2 %; 23.52 to 14.88 on sentences, −36.7 %), extending the OnHW WER finding to larger private vocabularies and long-context sentences. The CER result is different. REWI's CTC-trained BiLSTM remains stronger than the HWRFormer variants, which shows that the move to AR decoding changes the error profile rather than uniformly improving every metric. Headwise gating has a small observed CER advantage over elementwise on the private data (9.63 vs. 9.96 on words, 9.12 vs. 9.28 on sentences), while WER is essentially identical between the two.

4.3 Robust AR Training: Noise Injection

The migration study identifies where AR decoding helps and where it needs regularization. We next evaluate the five noise-injection modes defined in Sect. 3.2 at the operating point p_{ni}=0.15: Figure 2 compares the replacement modes with every other component held fixed, and Table 3 reports the uniform-mode results across datasets.

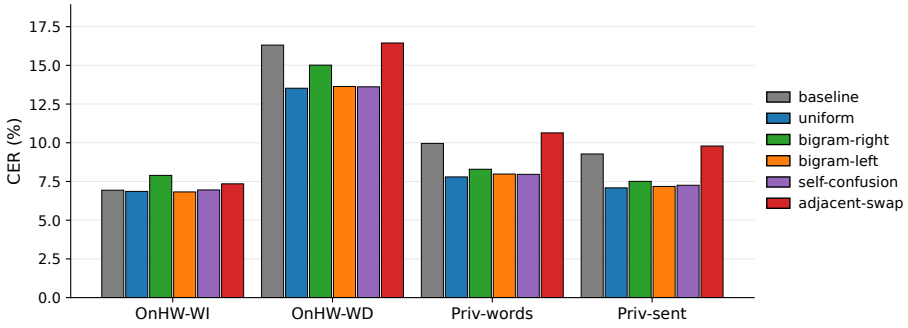

Fig. 2. Noise-injection mode comparison across the four word and sentence settings. 5-fold mean CER (%) at p_{ni}=0.15 on the elementwise-gated HWRFormer.

Mode comparison across replacement distributions Uniform is the most consistent default across the four settings, with the clearest CER gains on OnHW WD and both private subsets. Because the other substitution modes usually stay within about 1 pp CER, these means do not establish a superior structured replacement distribution. Adjacent-swap is the consistent negative case.

At the cross-dataset operating point $p_{\text{ni}} = 0.15$ used in Table 3, uniform noise changes OnHW WI only slightly (−1.2 %, 6.94 to 6.86), but it gives larger CER gains on OnHW WD (−17.1 %, 16.31 to 13.52), private words (−21.8 %, 9.96 to 7.79), and private sentences (−23.6 %, 9.28 to 7.09). This pattern is consistent with noise injection helping most when predicted prefixes drift farther from the teacher-forced prefixes seen during training.

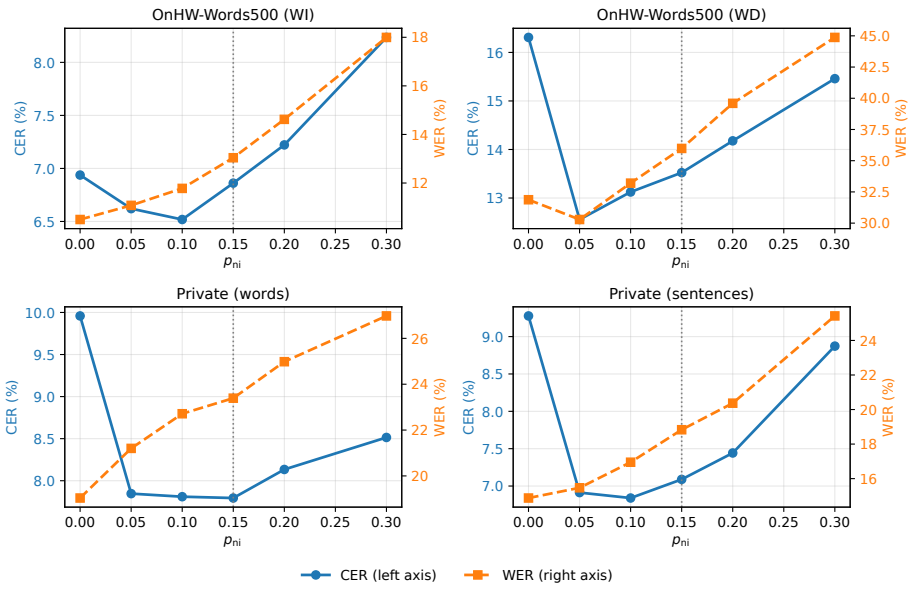

Fig. 3. Noise-rate sweep (uniform mode, 5-fold mean). CER is plotted on the left axis and WER on the right axis as functions of p_{ni}. The dotted vertical line marks the default $p_{ni} = 0.15$.

Noise-rate sweep We evaluate five noise rates (0.05, 0.10, 0.15, 0.20, 0.30) on all four word and sentence settings with full 5-fold cross-validation. Figure 3 traces CER and WER curves on OnHW WI, OnHW WD, private words, and private sentences.

1. CER is U-shaped in all four settings, with the empirical minimum at $p_{ni} = 0.10$ on OnHW WI and private sentences, at $p_{ni} = 0.05$ on OnHW WD, and at $p_{ni} = 0.15$ on private words. We use $p_{ni} = 0.15$ as a single cross-dataset default rather than as a per-dataset optimum; it stays within 1 pp of each CER minimum, and on private words it coincides with the optimum.
2. WER monotonically degrades with p_{ni} on OnHW WI, private words, and private sentences, while OnHW WD dips slightly at $p_{ni} = 0.05$ before rising. Higher noise rates can change an output into another plausible word and therefore hurt WER.

4.4 Hybrid CTC–AR Training

Table 3 shows that hybrid CTC–AR training is most useful on OnHW word recognition. On the WI split, hybrid alone gives the best WER and hybrid plus noise gives the best CER, so the auxiliary alignment objective yields a small observed gain over the AR recognizer. This pattern is stronger on WD. Hybrid improves WER, while hybrid plus noise gives the largest CER reduction.

Table 3. Cross-dataset effect of noise injection and hybrid training: 5-fold mean and across-fold standard deviation (STD) of CER and WER in %; lower is better. HWRFormer denotes the elementwise-gated AR variant from Table 1; "+" rows add the labeled training intervention on top of HWRFormer. Best mean per dataset and metric in **bold**.

Model	Metric	OnHW (WI)		OnHW (WD)		Priv. (words)		Priv. (sent.)	
		CER	WER	CER	WER	CER	WER	CER	WER
REWI [11]	Mean	7.30	15.16	14.81	44.77	9.39	31.82	**6.55**	23.52
	STD	6.85	10.05	3.84	7.84	0.59	0.69	0.71	2.27
HWRFormer	Mean	6.94	10.50	16.31	31.87	9.96	**19.04**	9.28	**14.88**
	STD	7.10	8.26	6.20	9.31	0.44	0.56	1.50	1.89
+ Noise	Mean	6.86	13.04	13.52	35.98	**7.79**	23.39	7.09	18.83
	STD	6.66	8.82	4.06	6.17	0.69	1.33	0.78	1.60
+ Hybrid	Mean	6.83	**10.17**	13.39	**27.42**	9.37	19.65	9.38	16.98
	STD	7.10	8.20	5.70	9.28	0.72	1.02	1.33	2.00
+ Hybrid + Noise	Mean	**6.70**	12.93	**11.49**	31.72	7.85	24.88	9.73	27.52
	STD	6.64	9.21	3.99	6.42	0.60	1.31	0.45	1.15

The private settings in the same table behave differently. Noise alone gives the best private-word CER, HWRFormer alone gives the best WER on both private subsets, and REWI remains strongest on private-sentence CER. Stacking hybrid and noise worsens private WER relative to either intervention alone, especially on sentences. Hybrid is therefore a targeted alignment regularizer for OnHW words rather than a universal default.

Direct exposure-bias probe Using the best-CER checkpoints underlying Table 3, the exposure-bias probe evaluates each HWRFormer recipe with and without teacher forcing; Table 4 reports the resulting CER values and gap reductions.

Removing teacher forcing exposes a sizeable prefix-drift gap for the AR-only HWRFormer: greedy autoregressive CER is 4.19–6.55 pp higher than teacher-forced CER, with the largest gap on private sentences. Relative to this AR-only exposure-bias gap, noise injection closes 61.1 % on OnHW WI, 67.4 % on OnHW WD, 72.9 % on private words, and 62.3 % on private sentences. Hybrid CTC–AR alone reduces the gap much less (3.1–34.8 %), which supports its interpretation as an encoder-side regularizer rather than a direct exposure-bias remedy. The combined recipe gives the largest reduction on three datasets, but remains close to noise alone on private sentences, suggesting that the two interventions partly overlap on the same exposure-bias axis. We therefore use Table 4 as a diagnostic of exposure-bias mitigation, while Table 3's greedy autoregressive metrics remain the primary recognition results.

Table 4. Teacher forcing (TF) vs. greedy autoregressive decoding for CER (%) on the same fold checkpoints. Within each method pair, the first row uses greedy autoregressive decoding with self-fed predictions, while the second row marked TF feeds the ground-truth prefix at every step and takes a per-position argmax. "Gap red." is the percentage reduction of the CER gap between greedy autoregressive and teacher-forced decoding relative to HWRFormer on the same dataset. All entries are recomputed in this paired diagnostic evaluator on the same best-CER checkpoints as Table 3. Best gap reduction per dataset in **bold**.

TF	Noise	Hybrid	OnHW (WI)		OnHW (WD)		Priv. (words)		Priv. (sent.)	
			CER	Gap red.	CER	Gap red.	CER	Gap red.	CER	Gap red.
			6.95	0.0%	16.31	0.0%	9.96	0.0%	9.28	0.0%
✓			2.76		10.91		5.23		2.73	
	✓		6.86	61.1%	13.52	67.4%	7.80	72.9%	7.09	**62.3%**
✓	✓		5.23		11.76		6.52		4.62	
		✓	6.83	3.1%	13.38	34.8%	9.37	10.4%	9.38	7.2%
✓		✓	2.77		9.86		5.13		3.30	
	✓	✓	6.70	**67.3%**	11.48	**78.7%**	7.85	**74.4%**	9.73	61.4%
✓	✓	✓	5.33		10.33		6.64		7.20	

λ_{ctc} selection Joint CTC–attention systems in speech recognition often use $\lambda_{\mathrm{ctc}} \approx 0.3$ to 0.6 [22,10]. We sweep $\lambda_{\mathrm{ctc}} \in \{0.1, 0.2, \ldots, 1.0\}$ on OnHW-words500 WI to select the value for IMU-based HWR (Fig. 4). The best setting is much lighter than the speech default: $\lambda_{\mathrm{ctc}} = 0.1$ is the only setting that beats AR-only on both CER and WER. For $\lambda_{\mathrm{ctc}} \geq 0.2$, downstream CER degrades while WER plateaus near the AR-only level, so the joint win disappears. We therefore adopt $\lambda_{\mathrm{ctc}} = 0.1$ throughout.

5 Discussion

HWRFormer changes the REWI error profile The controlled migration characterizes what changes when IMU-based HWR moves from recurrent CTC decoding to an AR Transformer. REWI uses CTC frame-level alignment, which favors character-level accuracy, whereas HWRFormer conditions each character on the generated prefix, giving the decoder a larger textual context and substantially reducing WER on the private subsets. The parameter-matched CNN-Transformer-CTC baseline does not reproduce this shift, so self-attention alone does not explain the result. HWRFormer also shows weaker character-level robustness when the predicted prefix diverges from the ground truth. Noise injection and hybrid CTC–AR training address this effect at complementary training locations: the former exposes the decoder to imperfect prefixes, while the latter supplies auxiliary encoder alignment supervision.

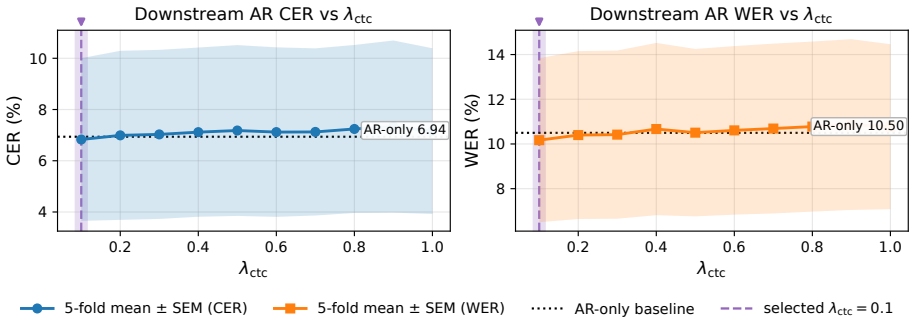

Fig. 4. Downstream metrics of the hybrid model vs. λ_{ctc} on OnHW WI. Curves show the 5-fold mean with standard error of the mean (SEM) bands; the selected value and AR-only reference are indicated in the plot.

When does noise injection help? Exposure bias predicts that an AR decoder trained under teacher-forced prefixes degrades most when its inference-time prefixes drift farthest from the training distribution. A single early prediction error is carried into every subsequent decoding step, so the gap grows with sequence length and with vocabulary breadth. The divergence across settings in Fig. 2 partly matches this prediction: noise injection has little effect on OnHW WI, where HWRFormer already achieves a low CER and leaves little room to improve. Its lower error rate and short sequences keep test-time prefixes closer to the teacher-forced training distribution, so prefix perturbation provides only a small signal. The gains are larger on OnHW WD, where HWRFormer has substantially higher CER, and on both private subsets, which use larger vocabularies, with the sentence subset additionally providing longer sequences. The strong WD result shows that sequence length and vocabulary breadth do not explain the pattern alone. The dataset settings also differ in writer overlap, hardware, and label distributions, so we read prefix drift as a consistent explanation rather than an isolated cause. Because the perturbation is applied only during training, noise injection changes robustness without adding inference cost.

The substitution modes cluster closely, which suggests that frequent prefix perturbation matters more than the exact replacement distribution. Adjacent-swap is the informative negative case. It perturbs the prefix by exchanging neighboring characters, which removes local order information without providing the substitution-style errors that resemble greedy autoregressive decoder mistakes. Its failure shows that not every prefix perturbation helps and makes a purely generic-noise explanation less likely. The direct probe in Table 4 gives the numerical check. On private sentences, noise injection reduces the teacher-forced versus greedy autoregressive CER gap by 62.3 %, while hybrid CTC–AR alone reduces it by only 7.2 %.

Hybrid CTC–AR is an encoder regularizer Hybrid CTC–AR training has its strongest observed effect on the OnHW word splits. On the WI split, the primary generalization test, it adds a modest alignment benefit on top of HWRFormer's word-level gain, and the stronger hybrid and hybrid-plus-noise gains on WD, where the writer style is familiar, are consistent with auxiliary alignment sharpening character-level evidence for familiar writing styles. Because the auxiliary CTC head acts only on the encoder during training, this reading is consistent with encoder-side regularization, although the WI/WD comparison does not identify the underlying bottleneck. On the private subsets, hybrid alone does not improve WER over HWRFormer and the hybrid-plus-noise combination degrades it further, whereas noise injection alone gives the broadest CER gains. This split is consistent with large-vocabulary and long-context inference depending more on prefix robustness than on sharper encoder evidence.

The upper end of the speech-style range, $\lambda_{\text{ctc}} = 0.6$, is too strong here, while $\lambda_{\text{ctc}} = 0.1$ supplies alignment pressure without letting the CTC objective dominate. The λ_{ctc} sweep shows that increasing the auxiliary loss weight does not track downstream recognition quality, which makes a "stronger CTC objective implies better recognition" explanation unlikely. The auxiliary head is therefore best read as an encoder regularizer rather than as an alternative recognizer.

Hybrid and noise injection are complementary, not additive The two interventions act on overlapping parts of the same robustness problem, so diminishing returns are expected, and stacking them is not additive under the evaluated settings: it lowers CER on OnHW but worsens private WER relative to either intervention alone. We therefore present the interventions as targeted alternatives rather than as a universal stacked recipe.

6 Conclusion

We introduced HWRFormer, the first AR-based model for IMU-based HWR, to test whether attention-based decoding can improve a modality still dominated by recurrent recognizers trained with CTC. The controlled migration shows a clear change in error profile: AR decoding improves word-level recognition but can make character accuracy more sensitive to prefix drift. Noise injection improves decoder-prefix robustness, whereas hybrid CTC–AR training provides training-only encoder supervision; their combined gains are not additive across datasets.

Future work includes quantifying the noise-injection mechanism beyond IMU handwriting and confidence-aware fallback decoding with HWRFormer's implicit language model.

Use of Generative AI. Generative AI tools supported language editing, proofreading, and consistency checks; the authors reviewed and verified all technical content, citations, and final wording.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Alemayoh, T.T., Shintani, M., Lee, J.H., Okamoto, S.: Deep-learning-based character recognition from handwriting motion data captured using IMU and force sensors. Sensors **22**(20), 7840 (2022). https://doi.org/10.3390/s22207840
2. Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N.: Scheduled sampling for sequence prediction with recurrent neural networks. In: Advances in Neural Information Processing Systems (NIPS). pp. 1171–1179 (2015), https://proceedings.neurips.cc/paper_files/paper/2015/file/e995f98d56967d946471af29d7bf99f1-Paper.pdf
3. Bluche, T., Louradour, J., Messina, R.: Scan, attend and read: End-to-end handwritten paragraph recognition with MDLSTM attention. In: Proceedings of the International Conference on Document Analysis and Recognition (ICDAR). pp. 1050–1055 (2017). https://doi.org/10.1109/ICDAR.2017.174
4. Bowman, S.R., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., Bengio, S.: Generating sentences from a continuous space. In: Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL). pp. 10–21. Association for Computational Linguistics, Berlin, Germany (2016). https://doi.org/10.18653/v1/K16-1002
5. Bublin, M., Werner, F., Kerschbaumer, A., Korak, G., Geyer, S., Rettinger, L., Schönthaler, E., Schmid-Kietreiber, M.: Handwriting evaluation using deep learning with SensoGrip. Sensors **23**(11), 5215 (2023). https://doi.org/10.3390/s23115215
6. Chan, W., Jaitly, N., Le, Q., Vinyals, O.: Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 4960–4964 (2016). https://doi.org/10.1109/ICASSP.2016.7472621
7. Ding, Y., Xue, Y.: A deep learning approach to writer identification using inertial sensor data of air-handwriting. IEICE Transactions on Information and Systems **E102.D**(10), 2059–2063 (2019). https://doi.org/10.1587/transinf.2019EDL8070
8. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning (ICML). pp. 369–376 (2006). https://doi.org/10.1145/1143844.1143891
9. Gulati, A., Qin, J., Chiu, C.C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., Pang, R.: Conformer: Convolution-augmented Transformer for speech recognition. In: Proceedings of Interspeech. pp. 5036–5040 (2020). https://doi.org/10.21437/Interspeech.2020-3015
10. Kim, S., Hori, T., Watanabe, S.: Joint CTC-attention based end-to-end speech recognition using multi-task learning. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 4835–4839 (2017). https://doi.org/10.1109/ICASSP.2017.7953075
11. Li, J., Hamann, T., Barth, J., Kämpf, P., Zanca, D., Eskofier, B.: Robust and efficient writer-independent IMU-based handwriting recognition. In: Sensor-Based Activity Recognition and Artificial Intelligence (iWOAR 2025). Lecture Notes in Computer Science, vol. 16292, pp. 261–286. Springer Nature Switzerland, Cham (2026). https://doi.org/10.1007/978-3-032-13312-0_16
12. Loshchilov, I., Hutter, F.: SGDR: Stochastic gradient descent with warm restarts. In: International Conference on Learning Representations (ICLR) (2017), https://openreview.net/forum?id=Skq89Scxx

13. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (ICLR) (2019), https://openreview.net/forum?id=Bkg6RiCqY7
14. Michael, J., Labahn, R., Grüning, T., Zöllner, J.: Evaluating sequence-to-sequence models for handwritten text recognition. In: Proceedings of the International Conference on Document Analysis and Recognition (ICDAR). pp. 1286–1293 (2019). https://doi.org/10.1109/ICDAR.2019.00208
15. Mihaylova, T., Martins, A.F.T.: Scheduled sampling for transformers. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop. pp. 351–356. Association for Computational Linguistics, Florence, Italy (2019). https://doi.org/10.18653/v1/P19-2049
16. Ott, F., Rügamer, D., Heublein, L., Hamann, T., Barth, J., Bischl, B., Mutschler, C.: Benchmarking online sequence-to-sequence and character-based handwriting recognition from IMU-enhanced pens. International Journal on Document Analysis and Recognition (IJDAR) **25**(4), 385–414 (2022). https://doi.org/10.1007/s10032-022-00415-6
17. Plamondon, R., Srihari, S.N.: Online and off-line handwriting recognition: A comprehensive survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(1), 63–84 (2000). https://doi.org/10.1109/34.824821
18. Qiu, Z., Wang, Z., Zheng, B., Huang, Z., Wen, K., Yang, S., Men, R., Yu, L., Huang, F., Huang, S., Liu, D., Zhou, J., Lin, J.: Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. In: Belgrave, D., Zhang, C., Lin, H., Pascanu, R., Koniusz, P., Ghassemi, M., Chen, N. (eds.) Advances in Neural Information Processing Systems. vol. 38, pp. 100092–100118. Curran Associates, Inc. (2025), https://proceedings.neurips.cc/paper_files/paper/2025/file/904e89bb4e632e75fb47f093b620b257-Paper-Conference.pdf
19. Shazeer, N.: GLU variants improve Transformer. arXiv preprint arXiv:2002.05202 (2020), https://arxiv.org/abs/2002.05202
20. Toffoli, S., Lunardini, F., Parati, M., Gallotta, M., De Maria, B., Longoni, L., Dell'Anna, M.E., Ferrante, S.: Spiral drawing analysis with a smart ink pen to identify Parkinson's disease fine motor deficits. Frontiers in Neurology **14**, 1093690 (2023). https://doi.org/10.3389/fneur.2023.1093690
21. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems (NeurIPS). vol. 30, pp. 6000–6010 (2017), https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html
22. Watanabe, S., Hori, T., Kim, S., Hershey, J.R., Hayashi, T.: Hybrid CTC/attention architecture for end-to-end speech recognition. IEEE Journal of Selected Topics in Signal Processing **11**(8), 1240–1253 (2017). https://doi.org/10.1109/JSTSP.2017.2763455
23. Wehbi, M., Hamann, T., Barth, J., Kaempf, P., Zanca, D., Eskofier, B.: Towards an IMU-based pen online handwriting recognizer. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) Document Analysis and Recognition – ICDAR 2021. Lecture Notes in Computer Science, vol. 12823, pp. 289–303. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-86334-0_19