

Adaptation Over Aggregation: Revisiting Pooling for Mamba-Based Cross-Domain ABSA

Hongzi Xu

Wenzhou-Kean University

Gaurav Gupta

ggupta@kean.edu

Kean University

Research Article

Keywords: Aspect-based sentiment analysis, cross-domain learning, Mamba, sequence pooling, pseudo-label adaptation

Posted Date: October 21st, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11049336/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Adaptation Over Aggregation: Revisiting Pooling for Mamba-Based Cross-Domain ABSA

Hongzi Xu^{1,2} and Gaurav Gupta^{1,2*}

¹Wenzhou-Kean University, 8 Daxue Rd, Ouhai, Wenzhou, Zhejiang, 325060, China.

²Kean University, 1000 Morris Avenue, Union, NJ, 07083, USA.

*Corresponding author(s). E-mail(s): ggupta@kean.edu;
Contributing authors: xuhongz@kean.edu;

Abstract

Cross-domain ABSA models must transfer sentiment knowledge without target labels. Mamba has no [CLS] token, so someone has to decide how the final hidden states become a sentence vector, and that decision is rarely examined. Under a transductive adaptation protocol, we benchmark Last-token, Mean, Max, and a scalar-gated Mean–Max Hybrid across two domain transfers, 14 configurations, and 5 seeds, with BERT reference models, ablations, significance tests, class-wise diagnostics, and efficiency profiles. The simplest rule wins: Last-token records the best Mamba Macro-F1 we observed (50.43 and 55.01), just ahead of the learned Hybrid gate (49.27 and 54.38), though the gap does not survive Holm correction. With Last-token pooling, pseudo-label adaptation lifts Mamba source-only Macro-F1 by 5.68 and 5.04 points, closing most of the distance to the BERT references, while calibration degrades in the process. Neutral and Conflict remain the weakest classes. At 64-token inputs, Mamba’s theoretical efficiency advantage does not appear: our implementation is roughly 4× slower at batch size 1 and ships a larger checkpoint than BERT. The practical message is that adaptation, not pooling sophistication, carries the transfer.

Keywords: Aspect-based sentiment analysis, cross-domain learning, Mamba, sequence pooling, pseudo-label adaptation

1 Introduction

A product review rarely has one mood. Aspect-based sentiment analysis (ABSA) handles this by assigning sentiment to specific targets—battery life, keyboard, or service—rather than a single document label. SemEval made Laptop and Restaurant the go-to testbeds [14]. Early LSTMs with aspect-conditioned attention [11, 20] and later BERT adaptations [5, 19, 22] progressively raised the bar.

Transferring these models across domains is harder. Reviewers talk about “fast” differently

for laptops (performance) than for restaurants (service). Domain adaptation theory frames this as a source–target discrepancy [1]; practitioners fight it with domain-adaptive pretraining [10] or adversarial learning [6]. Rietzler et al. further showed that domain-specific BERT adaptation improves cross-domain robustness [16]. Pseudo-labeling offers a label-light path [17], though it risks propagating errors—a cost we explicitly measure.

Mamba offers an efficient state-space backbone [4, 7, 8], but it lacks a [CLS] token. Suddenly, how we summarize the final hidden states becomes a

design choice: take the last token, average them, take the max, or learn a hybrid? Prior work on sentence embeddings shows that pooling choices materially affect what transfers [3, 15]. We treat this as an empirical question, not a chance to champion Hybrid pooling as universally superior.

Four research questions organize our analysis:

- **RQ1:** Can a source-only Mamba model transfer directly across Laptop and Restaurant, and how much does pseudo-label adaptation narrow its gap to BERT?
- **RQ2:** Which of Last-token, Mean, Max, and Hybrid pooling gives the strongest observed cross-domain performance under matched supervision?
- **RQ3:** Which components—pooling, pseudo-label adaptation, confidence weighting, and class weighting—drive the observed changes?
- **RQ4:** Are those changes statistically reliable, class-balanced, calibrated, and computationally efficient?

Concretely, we contribute (i) a controlled, five-seed comparison of four Mamba pooling strategies across two transfer directions; (ii) ablations that separate pooling, pseudo-label adaptation, confidence weighting, and class weighting within the tested Mamba setup; (iii) a multi-faceted evaluation spanning minority-class behavior, paired significance, calibration, and implementation-level efficiency; and (iv) a frank report of negative results—Last-token is the strongest tested Mamba pooling baseline by mean Macro-F1, the tested scalar-gated Hybrid does not establish consistent superiority, and our Mamba implementation exhibits approximately fourfold higher batch-1 latency than the reported BERT references at sequence length 64.

2 Methodology

2.1 Task definition and evaluation protocol

Let the labeled source set be $\mathcal{D}_s = \{(x_i^s, a_i^s, y_i^s)\}_{i=1}^{N_s}$, where x_i^s is a review sentence, a_i^s is an aspect term, and $y_i^s \in \{\text{Negative, Neutral, Positive, Conflict}\}$. The target set $\mathcal{D}_t = \{(x_j^t, a_j^t)\}_{j=1}^{N_t}$ is treated as unlabeled during training. Target labels are not used for

training, pseudo-label generation, hyperparameter selection, checkpoint selection, or calibration fitting.

The protocol is *transductive*: the unlabeled target instances used to produce pseudo-labels are the same instances on which final evaluation is performed after their labels are revealed. Consequently, the results should not be interpreted as an inductive evaluation on a separate, untouched target test set.

2.2 Input construction

Each aspect-level record contains a sentence, an aspect, and a polarity label. Both encoders receive the pair $(x, a) = (\text{sentence}, \text{aspect})$ with the sentence first and the aspect second. For BERT this corresponds to `[CLS] sentence [SEP] aspect [SEP]`. The implementation passes `input_ids` and `attention_mask`; although the tokenizer can produce segment identifiers, `token_type_ids` are not passed to the classifier. Mamba receives the same text pair through its own tokenizer without an additional prompt template. Inputs are truncated or padded to 64 tokens. In effect, BERT never sees an explicit sentence/aspect boundary through segment embeddings. This choice affects both the BERT reference models and the BERT teacher probabilities used for pseudo-labeling. Because the same teacher configuration is used for every Mamba pooling variant within a matched direction and seed, the within-Mamba pooling comparison retains matched supervision; however, the BERT results should be interpreted as implementation-specific references rather than as an exhaustively optimized segment-aware upper bound.

2.3 BERT teacher and pseudo-label adaptation

The teacher is `bert-base-uncased` implemented as `BertForSequenceClassification` with four output classes. All parameters are fine-tuned on the source training split. For each direction and random seed, the checkpoint with the highest source-validation Macro-F1 generates target-domain class probabilities

$$p_{jk} = \text{softmax}(z_j)_k, \quad (1)$$

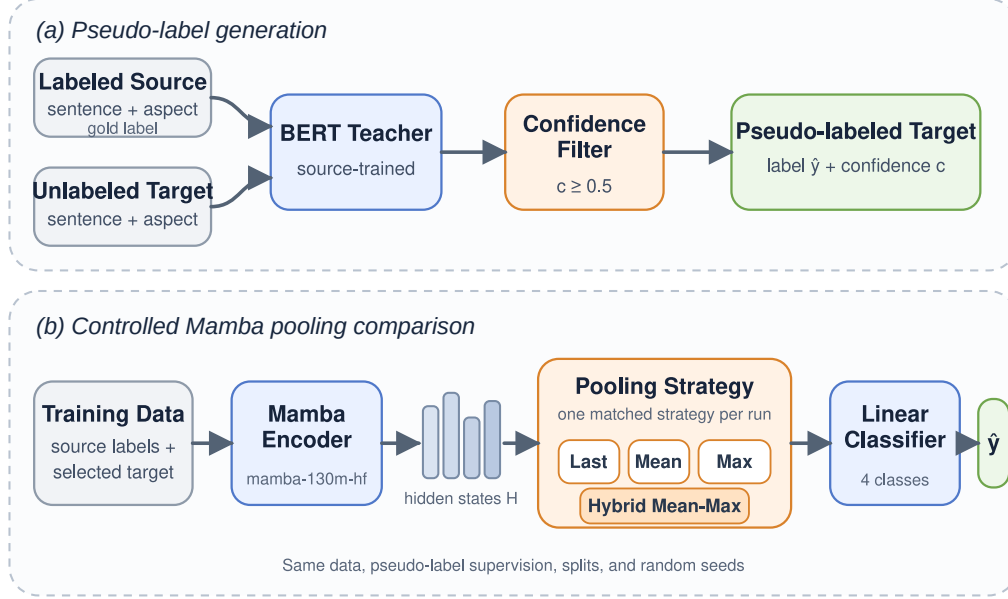


Fig. 1 Overview of the controlled cross-domain pipeline. A BERT teacher generates confidence-filtered pseudo-labels, while matched Mamba variants differ only in pooling and weighting

with pseudo-label and confidence

$$\hat{y}_j = \arg \max_k p_{jk}, \quad c_j = \max_k p_{jk}. \quad (2)$$

Only target instances satisfying $c_j \geq 0.5$ are retained. The number retained varies with the teacher seed: 3,097–3,623 for Laptop→Restaurant and 1,895–2,325 for Restaurant→Laptop. The threshold is fixed globally across transfer directions and seeds and is not selected using target labels; accordingly, no claim is made that 0.5 is an optimal cutoff or that the reported gains are insensitive to the threshold. The source-trained BERT teacher is kept fixed during pseudo-label generation so that all Mamba pooling variants receive matched supervision. Teacher-refresh and EMA-based teacher strategies are not compared in the present study.

For the unweighted pseudo-label configurations, source and pseudo-label losses enter with equal coefficients:

$$\mathcal{L} = \mathcal{L}_{\text{src}} + \mathcal{L}_{\text{pseudo}}. \quad (3)$$

The BERT pseudo-self-training baseline uses the same teacher-generated pseudo-labels to retrain BERT on the union of source and selected target instances. BERT-DANN combines the BERT

encoder with sentiment and domain classifiers and uses a fixed domain-loss coefficient of 1.0 [6]. We report this fixed-coefficient implementation as a reference point only; it is not used to support a general claim about the effectiveness of domain-adversarial adaptation.

2.4 Mamba encoder and pooling strategies

The student encoder is `state-spaces/mamba-130m-hf`, loaded through the Hugging Face `AutoModel` interface and fully fine-tuned. Let $H = [\mathbf{h}_1, \dots, \mathbf{h}_T] \in \mathbb{R}^{T \times d}$ be the final hidden sequence and $m_t \in \{0, 1\}$ the attention mask. The pooled vector is passed to a linear classifier $W_c \in \mathbb{R}^{4 \times d}$.

2.4.1 Last-token pooling

The representation is the final valid hidden state:

$$\mathbf{h}_{\text{last}} = \mathbf{h}_\ell, \quad \ell = \sum_{t=1}^T m_t. \quad (4)$$

2.4.2 Mean pooling

Padding states are excluded:

$$\mathbf{h}_{\text{mean}} = \frac{\sum_{t=1}^T m_t \mathbf{h}_t}{\sum_{t=1}^T m_t}. \quad (5)$$

2.4.3 Max pooling

For each hidden dimension r ,

$$\mathbf{h}_{\text{max}}[r] = \max_{t:m_t=1} \mathbf{h}_t[r]. \quad (6)$$

2.4.4 Hybrid pooling

Hybrid pooling uses a learnable scalar gate, not a concatenation of Last, Mean, and Max:

$$g = \sigma(W_g[\mathbf{h}_{\text{mean}}; \mathbf{h}_{\text{max}}] + b_g), \quad (7)$$

$$\mathbf{h}_{\text{hybrid}} = g\mathbf{h}_{\text{mean}} + (1 - g)\mathbf{h}_{\text{max}}, \quad (8)$$

where $W_g \in \mathbb{R}^{1 \times 2d}$, so every instance receives one scalar mixing coefficient.

2.5 Confidence gating and class weighting

For pseudo-labeled instances in the Full configuration, confidence scales the Hybrid gate logit:

$$g_j = \sigma(\text{clip}(c_j, 0.05, 1.0) (W_g[\mathbf{h}_{\text{mean}}; \mathbf{h}_{\text{max}}] + b_g)). \quad (9)$$

The pseudo-label loss is also confidence weighted:

$$\bar{c}_j = \text{clip}(c_j, 0.05, 1.0), \quad (10)$$

$$\mathcal{L}_{\text{pseudo}} = \frac{1}{N_p} \sum_{j=1}^{N_p} \bar{c}_j \text{CE}(f_{\theta}(x_j^t, a_j^t), \hat{y}_j). \quad (11)$$

Source-domain class weights are inverse-frequency weights capped at 5:

$$w_k = \min\left(\frac{N_s}{Cn_k}, 5.0\right), \quad C = 4, \quad (12)$$

where n_k is the source training count of class k . In the remainder of the paper, *Mamba Full* denotes Hybrid pooling, confidence mechanisms, class-weighted cross-entropy, and pseudo-label adaptation.

Figure 1 summarizes the matched experimental pipeline. The design isolates pooling and

weighting choices while holding the source split, target instances, teacher seed, and evaluation set fixed.

3 Experimental Setup

3.1 Data and transductive protocol

The project uses SemEval-2014 Task 4-style aspect-level sentiment data derived into JSON records. A sentence containing multiple aspect terms is expanded into multiple aspect-level instances. Table 1 reports the domain sizes and source splits. The source domain is divided with a stratified 90/10 split using the fixed split seed 20260803. When a domain serves as the target, its full instance set is used as unlabeled adaptation data and subsequently evaluated with the hidden labels.

Table 1 Aspect-level data and source-domain split

Domain	Total	Source		Target
		Train	Validation	
Laptop	2,358	2,122	236	2,358
Restaurant	3,693	3,323	370	3,693

Table 2 Fixed target-evaluation support by sentiment class

Direction	Negative	Neutral	Positive	Conflict
Laptop→Restaurant	805	633	2,164	91
Restaurant→Laptop	866	460	987	45

The target test supports in Table 2 reveal substantial class imbalance, particularly for Conflict. This motivates reporting Macro-F1 and class-wise F1 in addition to Accuracy and Weighted-F1.

3.2 Training configuration

All models are optimized with AdamW [13], learning rate 2×10^{-5} , weight decay 0.01, and no scheduler or warm-up. Training runs for at least 3 and at most 10 epochs with early-stopping

patience 3. Source-validation Macro-F1 selects the checkpoint. BERT uses batch size 8; Mamba uses batch size 2; data-loader workers are 0; no source or target sample limit is applied. The five experiment seeds are 13, 21, 42, 87, and 2026. Training and efficiency measurements use an NVIDIA GeForce RTX 4090 with PyTorch 2.10.0+cu128. The training batch sizes are not harmonized across architectures. Therefore, BERT–Mamba performance differences are descriptive comparisons under the reported implementations rather than strictly architecture-isolated effects. By contrast, the Mamba pooling variants share the same training batch size and optimization setup, so the controlled pooling comparison is not confounded by an across-pooling batch-size change.

The formal evidence contains 140 completed runs: 14 configurations, two transfer directions, and five matched seeds. Reported values are mean $\pm$ sample standard deviation with ddof = 1.

3.3 Metrics, significance, calibration, and efficiency

We report Accuracy, Macro-F1, and Weighted-F1. Macro-F1 weights each class equally and is therefore less dominated by the majority classes than Accuracy or Weighted-F1 [18]. Statistical comparisons pair methods by identical direction and seed. We use a paired *t*-test, a two-sided exact Wilcoxon signed-rank test [21], 95% confidence intervals for the paired mean difference, paired standardized effect size d_z , and Holm correction across planned comparisons [12]. With only five pairs, the minimum attainable exact two-sided Wilcoxon *p*-value is 0.0625; non-significance therefore must not be interpreted as equivalence.

Calibration is measured from raw checkpoint probabilities using negative log-likelihood (NLL), multiclass Brier score [2], expected calibration error (ECE) over 15 equal-width top-label confidence bins, maximum calibration error (MCE), and mean confidence, following the modern calibration literature [9]. No target-label temperature fitting is performed. Saved artifacts permit top-label reliability analysis but not per-class ECE reconstruction.

Efficiency uses seed-42 checkpoints, batch sizes 1 and 8, 10 warm-up iterations, 50 timed batches, and CUDA synchronization. GPU memory is

checked before launch, but continuous GPU exclusivity is not guaranteed; these results are controlled single-machine measurements rather than hardware-independent benchmarks.

4 Results

4.1 RQ1: Overall cross-domain performance

Table 3 reports the nine principal methods. In Laptop→Restaurant, BERT Source-only reaches 51.94 ± 1.40 Macro-F1, whereas Mamba Source-only reaches 44.75 ± 3.13 , a 7.19-point gap. In Restaurant→Laptop, the corresponding values are 55.18 ± 3.12 and 49.97 ± 6.88 , a 5.21-point gap. The Mamba backbone alone, then, buys no transfer advantage.

Pseudo-label adaptation substantially changes the comparison. Mamba with Last-token pooling improves over Mamba Source-only by 5.68 points in Laptop→Restaurant and 5.04 points in Restaurant→Laptop, reaching 50.43 ± 2.31 and 55.01 ± 3.37 . It consequently approaches the reported BERT + Pseudo reference, trailing by only 0.33 and 0.06 points. By contrast, BERT pseudo-self-training does not improve mean Macro-F1 over BERT Source-only in either direction. Under the tested implementations, pseudo-label adaptation therefore yields a larger observed mean gain for Mamba Last than for BERT pseudo-self-training; this comparison should not be interpreted as an architecture-isolated causal effect.

The accuracy–Macro-F1 plane in Fig. 2 shows that methods clustered near 70–73% Accuracy can differ materially in Macro-F1. The collapsed BERT-DANN baseline is the exception, with low means and, in Laptop→Restaurant, very high seed variance. This finding concerns the matched implementation and does not establish that domain-adversarial training is generally ineffective.

4.2 RQ2: Pooling strategy comparison

Table 4 controls the source data, BERT-generated pseudo-labels, target instances, and random seeds while changing only Mamba pooling. Last-token pooling obtains the highest

Table 3 Main cross-domain results. Values are mean $\pm$ sample standard deviation (percentage points) over five matched seeds. Bold marks the highest Macro-F1 in each direction; italics mark the second highest

Method	Laptop→Restaurant			Restaurant→Laptop		
	Accuracy	Macro-F1	Weighted-F1	Accuracy	Macro-F1	Weighted-F1
BERT Source-only	70.84 $\pm$ 1.70	51.94$\pm$1.40	70.34 $\pm$ 1.11	72.67 $\pm$ 1.24	55.18$\pm$3.12	71.54 $\pm$ 2.50
BERT + Pseudo	71.28 $\pm$ 1.94	<i>50.76$\pm$1.68</i>	70.46 $\pm$ 1.30	73.17 $\pm$ 1.80	<i>55.07$\pm$3.27</i>	71.85 $\pm$ 2.80
BERT-DANN	38.60 $\pm$ 22.07	17.03 $\pm$ 10.48	26.47 $\pm$ 23.52	41.43 $\pm$ 1.76	15.73 $\pm$ 0.31	25.40 $\pm$ 1.02
Mamba Source-only	71.71 $\pm$ 1.03	44.75 $\pm$ 3.13	68.54 $\pm$ 2.13	68.13 $\pm$ 6.63	49.97 $\pm$ 6.88	66.62 $\pm$ 5.52
Mamba + Pseudo (Last)	70.73 $\pm$ 1.59	50.43 $\pm$ 2.31	69.95 $\pm$ 1.12	73.09 $\pm$ 1.12	55.01 $\pm$ 3.37	71.67 $\pm$ 2.29
Mamba + Pseudo (Mean)	70.30 $\pm$ 1.72	49.22 $\pm$ 0.43	69.27 $\pm$ 0.74	72.82 $\pm$ 1.71	54.61 $\pm$ 3.10	71.37 $\pm$ 2.45
Mamba + Pseudo (Max)	69.66 $\pm$ 2.86	47.94 $\pm$ 2.53	68.70 $\pm$ 1.63	72.56 $\pm$ 1.66	54.10 $\pm$ 3.10	71.10 $\pm$ 2.39
Mamba + Pseudo (Hybrid)	70.78 $\pm$ 1.63	49.27 $\pm$ 0.79	69.59 $\pm$ 0.77	72.78 $\pm$ 1.28	54.38 $\pm$ 3.15	71.41 $\pm$ 2.17
Mamba Full (Hybrid + Conf. + CW)	70.40 $\pm$ 1.81	50.12 $\pm$ 2.63	69.70 $\pm$ 1.50	72.38 $\pm$ 1.70	53.98 $\pm$ 3.43	71.19 $\pm$ 2.87

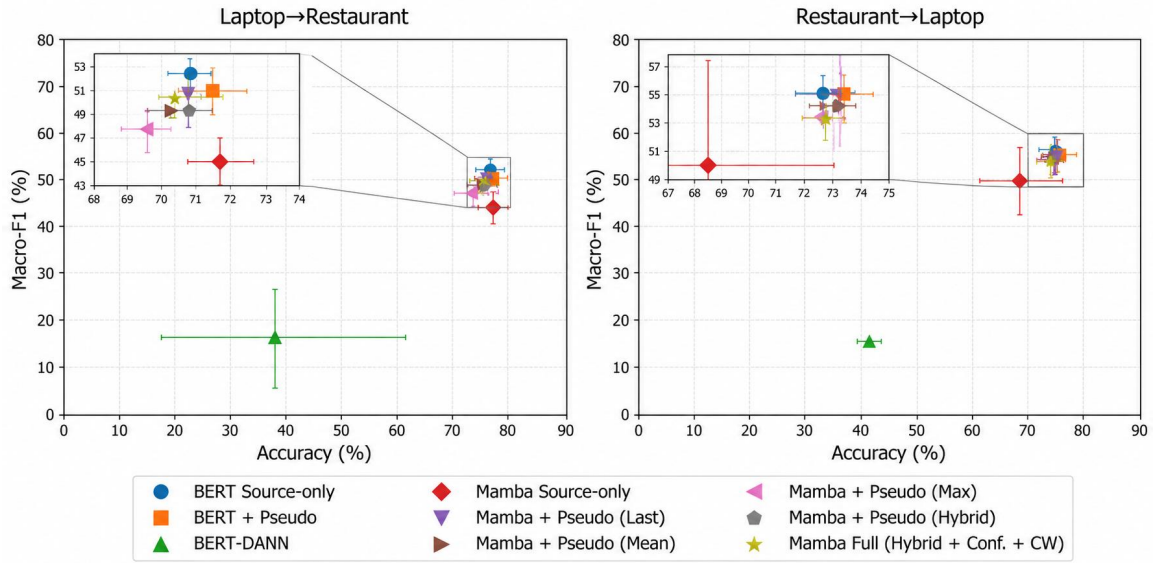


Fig. 2 Accuracy–Macro-F1 trade-off with one-sample-standard-deviation whiskers across matched seeds

observed Macro-F1 in both directions: 50.43 $\pm$ 2.31 in Laptop→Restaurant and 55.01 $\pm$ 3.37 in Restaurant→Laptop. Hybrid trails Last by 1.16 and 0.63 points, respectively. Mean is close to Hybrid, whereas Max is weakest in Laptop→Restaurant.

The error bars in Fig. 3 overlap substantially, especially in Restaurant→Laptop. The paired tests in Sect. 4.5 confirm that the observed mean ordering does not justify a universal or statistically significant superiority claim for Last or Hybrid. The supported conclusion is narrower: Last-token is the strongest tested Mamba pooling baseline by mean Macro-F1 under this protocol.

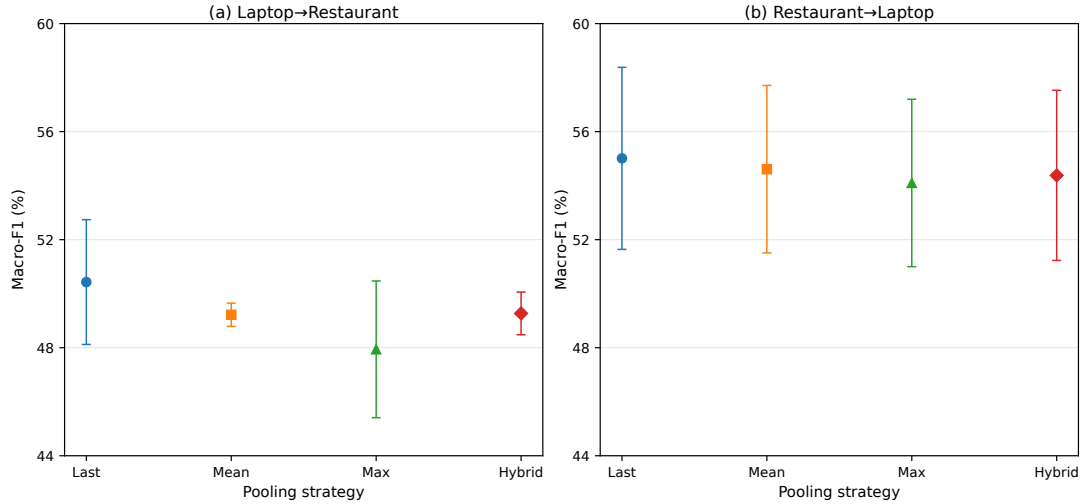
4.3 RQ3: Component and ablation analysis

Table 5 reports the complete factor study. Source-only to Pseudo-Last is an adaptation comparison, not a one-factor Full-model ablation. The strict single-factor ablations are: replacing Hybrid with Last while retaining confidence and class weighting; removing confidence while retaining Hybrid and class weighting; and removing class weighting while retaining Hybrid and confidence.

Replacing Hybrid with Last while keeping confidence and class weighting improves Macro-F1 by 0.88 points in Laptop→Restaurant and 1.37

Table 4 Controlled Mamba pooling comparison under identical pseudo-label supervision

Pooling	Laptop→Restaurant		Restaurant→Laptop	
	Accuracy	Macro-F1	Accuracy	Macro-F1
Last	70.73±1.59	50.43±2.31	73.09±1.12	55.01±3.37
Mean	70.30±1.72	49.22±0.43	72.82±1.71	54.61±3.10
Max	69.66±2.86	47.94±2.53	72.56±1.66	54.10±3.10
Hybrid	70.78±1.63	49.27±0.79	72.78±1.28	54.38±3.15

**Fig. 3** Controlled comparison of Last-token, Mean, Max, and Hybrid pooling under identical BERT-generated pseudo-label supervision

points in Restaurant→Laptop. Removing confidence from Full changes Macro-F1 by +0.60 and +0.09; removing class weighting changes it by −0.25 and +0.35. Component contributions flip sign with direction. The strongest factor configurations use Last pooling rather than Hybrid: Last + CW in Laptop→Restaurant and Last + Conf. + CW in Restaurant→Laptop. Figure 4 visualizes these changes relative to Full and reinforces that adaptation, not Hybrid aggregation alone, explains the largest recovery from source-only transfer.

4.4 RQ4a: Class-wise behavior

Table 6 reports five representative methods. Negative and Positive F1 are consistently much higher than Neutral and Conflict F1. In Laptop→Restaurant, Mamba Source-only obtains 0 Conflict F1 and only 28.71±13.30 Neutral F1, explaining why its Accuracy remains high while its Macro-F1 is low. Pseudo-label adaptation raises

those values to 12.63±8.40 and 40.52±4.41 for Mamba Last, but the minority classes remain the bottleneck. Conflict is particularly unstable because the fixed target sets contain only 91 and 45 instances.

Figure 5 shows the same imbalance graphically. The Accuracy–Macro-F1 gap is evidence of uneven class-wise performance, not by itself evidence of miscalibration; calibration is evaluated separately below. Δ_{Full} values are computed from unrounded per-seed means and may differ from the displayed rounded means by 0.01.

4.5 RQ4b: Paired statistical significance

Table 7 presents the comparisons most directly connected to the pooling and architecture claims. None of the Hybrid-versus-fixed-pooling comparisons is significant after Holm correction. Mamba Full also does not significantly outperform BERT

Table 5 Complete Mamba factor and ablation study. Δ Full is the Macro-F1 change relative to Mamba Full within the same direction

Configuration	Laptop→Restaurant			Restaurant→Laptop		
	Accuracy	Macro-F1	Δ Full	Accuracy	Macro-F1	Δ Full
Mamba Source-only	71.71 $\pm$ 1.03	44.75 $\pm$ 3.13	-5.37	68.13 $\pm$ 6.63	49.97 $\pm$ 6.88	-4.01
Mamba + Pseudo (Last)	70.73 $\pm$ 1.59	50.43 $\pm$ 2.31	+0.32	73.09 $\pm$ 1.12	55.01 $\pm$ 3.37	+1.03
Mamba + Pseudo (Mean)	70.30 $\pm$ 1.72	49.22 $\pm$ 0.43	-0.90	72.82 $\pm$ 1.71	54.61 $\pm$ 3.10	+0.63
Mamba + Pseudo (Max)	69.66 $\pm$ 2.86	47.94 $\pm$ 2.53	-2.17	72.56 $\pm$ 1.66	54.10 $\pm$ 3.10	+0.12
Mamba + Pseudo (Hybrid)	70.78 $\pm$ 1.63	49.27 $\pm$ 0.79	-0.85	72.78 $\pm$ 1.28	54.38 $\pm$ 3.15	+0.40
Mamba + Pseudo (Last + CW)	70.63 $\pm$ 1.85	51.03$\pm$2.06	+0.92	73.04 $\pm$ 1.40	55.21 $\pm$ 3.15	+1.23
Mamba + Pseudo (Last + Conf.)	71.01 $\pm$ 1.84	50.38 $\pm$ 3.15	+0.27	73.06 $\pm$ 1.32	55.21 $\pm$ 3.10	+1.23
Mamba + Pseudo (Last + Conf. + CW)	70.64 $\pm$ 2.00	51.00 $\pm$ 2.63	+0.88	73.07 $\pm$ 1.44	55.35$\pm$3.10	+1.37
Mamba + Pseudo (Hybrid + Conf.)	71.17 $\pm$ 1.64	49.86 $\pm$ 1.42	-0.25	72.90 $\pm$ 1.10	54.33 $\pm$ 2.94	+0.35
Mamba + Pseudo (Hybrid + CW)	70.30 $\pm$ 1.85	50.72 $\pm$ 2.00	+0.60	72.38 $\pm$ 1.34	54.07 $\pm$ 3.25	+0.09
Mamba Full (Hybrid + Conf. + CW)	70.40 $\pm$ 1.81	50.12 $\pm$ 2.63	0.00	72.38 $\pm$ 1.70	53.98 $\pm$ 3.43	0.00

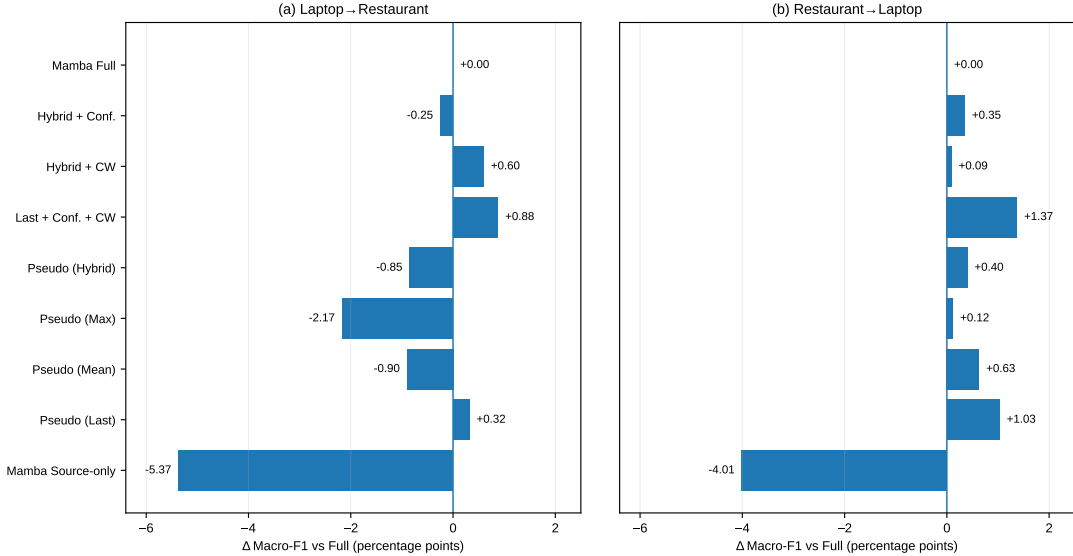


Fig. 4 Macro-F1 changes relative to the Full Mamba configuration; positive bars outperform Full within the same transfer direction

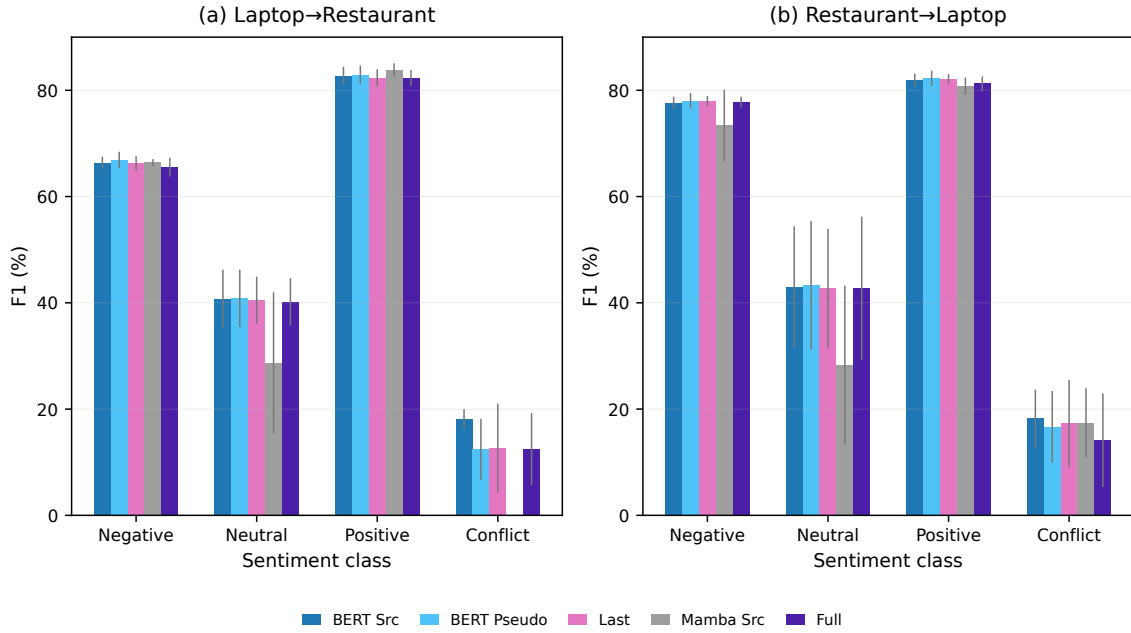
Source-only or BERT + Pseudo. The only Holm-significant comparisons in the complete testing family involve the collapsed BERT-DANN baseline. Accordingly, the evidence supports competitiveness after adaptation, not Mamba superiority, and identifies Last as the highest observed pooling mean without proving universal dominance.

4.6 RQ4c: Calibration

Table 8 reveals a consistent adaptation-calibration trade-off. In Laptop→Restaurant, BERT pseudo-self-training raises ECE from 14.61 $\pm$ 5.07% to 25.47 $\pm$ 3.95% and NLL from 0.927 $\pm$ 0.134 to 1.687 $\pm$ 0.554. Mamba Full likewise raises ECE from 14.36 $\pm$ 8.40% to 23.33 $\pm$ 4.82% and NLL from 1.105 $\pm$ 0.304 to 1.602 $\pm$ 0.581. The same direction appears in Restaurant→Laptop.

Table 6 Per-class F1 for representative methods. Values are mean $\pm$ sample standard deviation

Direction	Method	Negative	Neutral	Positive	Conflict
Laptop→Restaurant	BERT Source-only	66.26 $\pm$ 1.23	40.78 $\pm$ 5.46	82.71 $\pm$ 1.71	18.02 $\pm$ 1.96
	BERT + Pseudo	66.88 $\pm$ 1.55	40.85 $\pm$ 5.38	82.90 $\pm$ 1.76	12.43 $\pm$ 5.72
	Mamba Source-only	66.41 $\pm$ 0.64	28.71 $\pm$ 13.30	83.87 $\pm$ 1.20	0.00 $\pm$ 0.00
	Mamba + Pseudo (Last)	66.21 $\pm$ 1.38	40.52 $\pm$ 4.41	82.37 $\pm$ 1.60	12.63 $\pm$ 8.40
	Mamba Full	65.51 $\pm$ 1.77	40.18 $\pm$ 4.47	82.31 $\pm$ 1.49	12.47 $\pm$ 6.77
Restaurant→Laptop	BERT Source-only	77.60 $\pm$ 1.21	42.96 $\pm$ 11.44	81.96 $\pm$ 1.18	18.20 $\pm$ 5.45
	BERT + Pseudo	78.03 $\pm$ 1.45	43.31 $\pm$ 12.08	82.25 $\pm$ 1.43	16.67 $\pm$ 6.73
	Mamba Source-only	73.38 $\pm$ 6.71	28.28 $\pm$ 14.95	80.81 $\pm$ 1.59	17.42 $\pm$ 6.52
	Mamba + Pseudo (Last)	77.97 $\pm$ 0.98	42.68 $\pm$ 11.23	82.13 $\pm$ 0.92	17.28 $\pm$ 8.24
	Mamba Full	77.79 $\pm$ 1.03	42.72 $\pm$ 13.49	81.27 $\pm$ 1.35	14.14 $\pm$ 8.80

**Fig. 5** Per-class F1 for five representative methods, revealing persistent weakness and seed variability for Neutral and Conflict

Adapted models also produce very high mean confidence: 96.73–98.61% for BERT + Pseudo and 93.68–95.73% for Mamba Full.

The reliability curves in Fig. 6 lie predominantly below the diagonal at high confidence, consistent with overconfidence. Because only pooled top-label bins were saved, this study does not claim per-class calibration estimates.

4.7 RQ4d: Computational efficiency

Table 9 shows that the present Mamba implementation is not more efficient in this short-sequence ABSA setting. It contains 129.14M parameters and a 492.7 MB checkpoint, compared with 109.49M and 417.7 MB for the main BERT models. At batch size one, BERT requires approximately 3.45–3.56 ms for the main BERT models (3.71–3.86 ms for BERT-DANN), whereas Mamba requires approximately 14.0–14.2 ms. At

Table 7 Selected seed-matched Macro-F1 significance tests. Differences are $A - B$ in percentage points. Holm p values are adjusted within the planned comparison family

Direction	Comparison ($A - B$)	Δ	95% CI	Holm p	Wilcoxon p	d_z
Laptop→Restaurant	Hybrid – Last	−1.16	[−3.69, 1.36]	1.0000	0.3125	−0.57
Laptop→Restaurant	Hybrid – Mean	+0.05	[−0.68, 0.78]	1.0000	1.0000	+0.09
Laptop→Restaurant	Hybrid – Max	+1.33	[−1.95, 4.60]	1.0000	0.3125	+0.50
Laptop→Restaurant	Full – BERT Source-only	−1.83	[−4.47, 0.81]	0.7497	0.0625	−0.86
Laptop→Restaurant	Full – Mamba Source-only	+5.37	[−0.36, 11.10]	0.4197	0.0625	+1.16
Restaurant→Laptop	Hybrid – Last	−0.64	[−1.73, 0.46]	1.0000	0.1875	−0.72
Restaurant→Laptop	Hybrid – Mean	−0.24	[−1.35, 0.87]	1.0000	0.6250	−0.26
Restaurant→Laptop	Hybrid – Max	+0.27	[−0.60, 1.15]	1.0000	0.8125	+0.39
Restaurant→Laptop	Full – BERT Source-only	−1.20	[−3.45, 1.05]	1.0000	0.3125	−0.66
Restaurant→Laptop	Full – Mamba Source-only	+4.01	[−7.01, 15.02]	1.0000	0.4375	+0.45

Table 8 Calibration of five representative methods. ECE uses 15 equal-width top-label confidence bins

Direction	Method	NLL	Brier	ECE-15 (%)	MCE (%)	Mean confidence (%)
Laptop→Restaurant	BERT Source-only	0.927±0.134	0.443±0.029	14.61±5.07	29.51±9.16	85.41±5.83
	BERT + Pseudo	1.687±0.554	0.535±0.053	25.47±3.95	47.50±7.65	96.73±3.78
	BERT-DANN	2.230±1.307	1.031±0.417	45.58±27.54	56.14±15.60	74.80±20.40
	Mamba Source-only	1.105±0.304	0.453±0.037	14.36±8.40	24.76±14.16	85.33±10.12
	Mamba Full	1.602±0.581	0.529±0.032	23.33±4.82	43.11±15.19	93.68±6.11
Restaurant→Laptop	BERT Source-only	0.995±0.231	0.436±0.050	15.60±7.18	33.50±16.39	86.77±9.04
	BERT + Pseudo	1.702±0.148	0.520±0.040	25.52±2.24	56.19±10.60	98.61±0.57
	BERT-DANN	2.060±0.987	0.910±0.230	35.49±24.13	57.96±19.14	76.34±26.33
	Mamba Source-only	1.206±0.188	0.496±0.073	18.32±3.40	40.91±14.83	86.40±5.20
	Mamba Full	1.513±0.491	0.506±0.039	23.45±3.07	42.73±7.82	95.73±3.28

batch size eight, BERT processes about 1,913–1,941 samples/s, compared with about 506–528 samples/s for Mamba.

Figure 7 places Macro-F1 against batch-1 latency and model size. Mamba’s theoretical attraction for long sequences should not be conflated with an implementation-level speed advantage at a maximum length of 64. Kernel overheads, batch size, checkpoint size, and the absence of a long-context stress test all matter in interpreting this result.

5 Discussion

5.1 Why does last-token pooling form a strong baseline?

Why would the last token be enough? Mamba’s selective scan rolls the sequence into a single running state, so by construction the final valid position has already seen everything before it.

Mean and Max discard that ordering; Last-token keeps it for free. We offer this as intuition, not as a mechanism we have verified: we do not map final valid positions back to token categories, and we do not test whether Last-token performance tracks aspect-final positioning.

Mean pooling treats all valid states uniformly, while Max pooling selects dimension-wise extremes that may amplify noisy or domain-specific activations. The scalar Hybrid gate can interpolate between those two statistics but cannot model dimension-specific or token-specific mixing. This limited parameterization may be insufficient to offset pseudo-label noise and class imbalance. Accordingly, the lack of consistent superiority observed here applies only to the tested scalar-gated Mean–Max Hybrid; no conclusion is drawn about attention pooling, learnable token-weighted sums, or token- and dimension-wise gating.

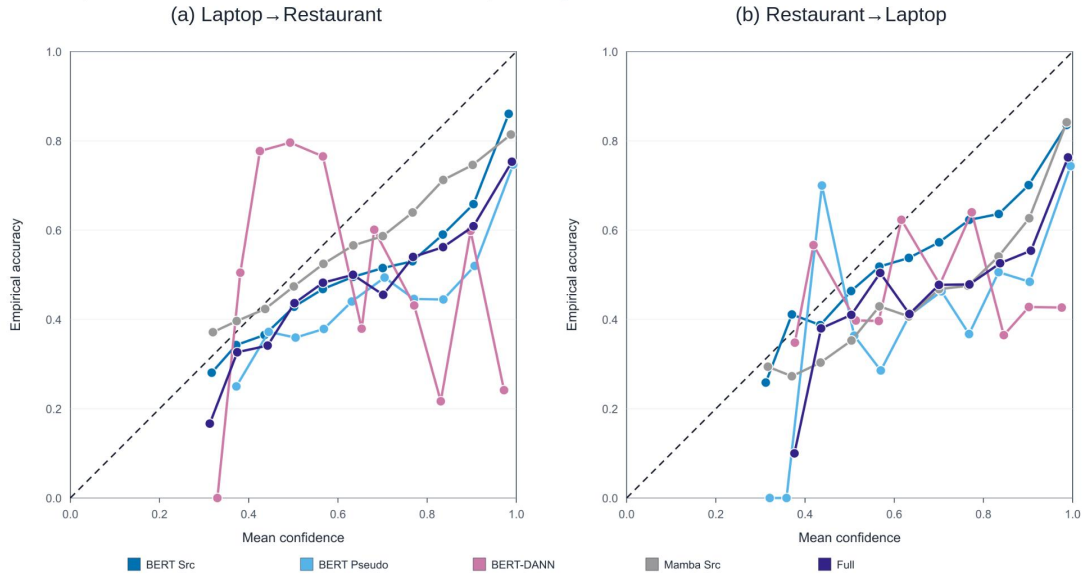


Fig. 6 Per-class F1 scores of representative methods for cross-domain sentiment classification in the Laptop→Restaurant and Restaurant→Laptop directions. Bars show mean F1 scores, and error bars indicate sample standard deviations. “Last” denotes Mamba + Pseudo (Last).

Table 9 Controlled training and inference efficiency. Inference uses seed-42 checkpoints; training time is mean $\pm$ sample standard deviation over five seeds where available. NR denotes not recorded

Direction	Method	Params	Check-point (MB)	Train (min)	BS1 latency (ms/sample)	BS8 throughput (samples/s)	BS8 peak (MB)
Laptop→Restaurant	BERT-DANN	109.78M	418.8	3.95 $\pm$ 0.75	3.707	1,927.4	449
	BERT + Pseudo	109.49M	417.7	6.54 $\pm$ 2.00	3.561	1,941.0	447
	BERT Source-only	109.49M	417.7	1.89 $\pm$ 0.55	3.497	1,913.0	446
	Mamba Full	129.14M	492.7	21.40 $\pm$ 3.94	14.176	516.9	1,051
	Mamba Source-only	129.14M	492.7	8.82 $\pm$ 1.33	14.014	506.3	558
Restaurant→Laptop	BERT-DANN	109.78M	418.8	4.68 $\pm$ 1.25	3.857	1,930.4	942
	BERT + Pseudo	109.49M	417.7	6.54 $\pm$ 1.01	3.453	1,922.2	940
	BERT Source-only	109.49M	417.7	2.49 $\pm$ 1.05	3.478	1,924.7	939
	Mamba Full	129.14M	492.7	23.02 $\pm$ 8.78	14.032	523.4	1,545
	Mamba Source-only	129.14M	492.7	13.52 $\pm$ 2.65	14.003	527.6	1,051

5.2 Adaptation matters more than pooling complexity

The most stable change is the 4–6 point recovery from Mamba Source-only to pseudo-label-adapted variants. This suggests that exposure to target-domain sentence–aspect pairs is more consequential than replacing one reasonable pooling

statistic with a more complex mixture. However, pseudo-label adaptation does not universally improve every model: BERT pseudo-self-training slightly reduces mean Macro-F1 relative to BERT Source-only. The benefit depends on the interaction among teacher errors, student architecture, class distribution, and transfer direction.

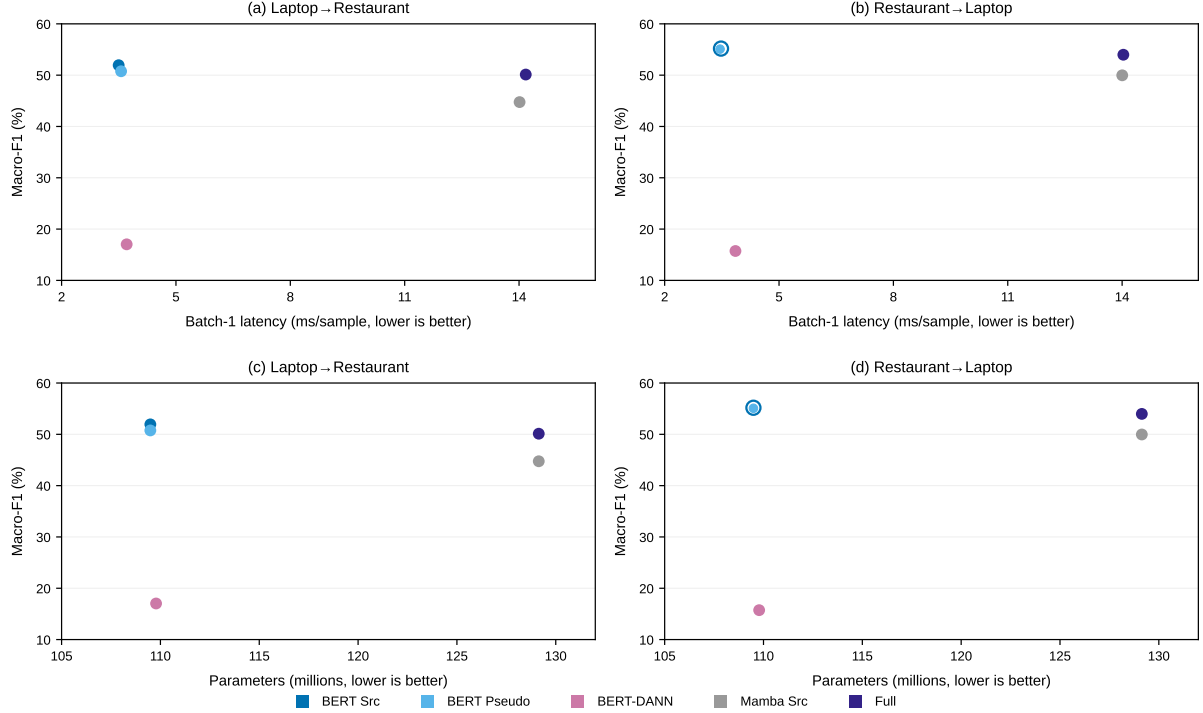


Fig. 7 Macro-F1 versus batch-1 latency and parameter count for representative methods. BERT Source-only and BERT + Pseudo nearly overlap in panels (b) and (d); different marker styles are used to make both points visible.

5.3 Aggregate performance hides reliability problems

Accuracy and Weighted-F1 are dominated by Positive and Negative instances. Macro-F1 and class-wise analysis expose persistent Neutral and Conflict weakness. The calibration analysis adds a distinct warning: adapted models become more confident without corresponding reliability gains. A practical cross-domain system should therefore consider confidence calibration, class-aware pseudo-label selection, or uncertainty-aware rejection rather than optimizing Accuracy alone.

5.4 Theoretical and measured efficiency are different claims

Mamba’s linear sequence-length scaling is a model-level property demonstrated in long-sequence settings [7]. The present workload uses 64-token inputs, a 130M-parameter checkpoint, small Mamba batches, and a particular Hugging Face implementation. Under these conditions, the reported BERT references are smaller

and faster. Because training batch sizes differ across architectures, training-time measurements are implementation-level profiles rather than batch-normalized architecture comparisons. The batch-1 and batch-8 inference measurements are more directly matched with respect to inference batch size, but they remain single-hardware, implementation-specific measurements. The appropriate conclusion is therefore not that Mamba is generally inefficient, but that its theoretical scaling advantage does not materialize automatically in this short-sequence setting.

6 Limitations

Protocol.. The evaluation is transductive: the same unlabeled target instances feed pseudo-label adaptation and the final labeled evaluation. Our numbers characterize that setting and say nothing about inductive generalization to a held-out target set.

Implementation.. We do not pass `token.type_ids` to BERT, which affects both

the BERT references and the teacher probabilities. Cross-architecture comparisons are therefore implementation-specific. The within-Mamba comparison is unaffected; every pooling variant sees pseudo-labels from the same teacher at the same seed; however, BERT and Mamba also train at different batch sizes, so architecture-level timing claims are profiles, not controlled contrasts.

Pseudo-labeling.. A fixed global threshold of 0.5 and a frozen source-trained teacher. We ran no threshold sweep, and we did not compare EMA teachers, teacher refresh, or class-adaptive selection.

Scope.. Two domains, one BERT backbone, one Mamba backbone, four pooling rules. Five seeds bound the exact two-sided Wilcoxon p at 0.0625, so nothing here should be read as an equivalence result. Calibration is diagnostic only: we saved top-label bins, not per-class ones, and applied no temperature scaling. Efficiency comes from one GPU at length 64 and never touches the long-context regime the architecture was designed for. The BERT-DANN collapse reflects our fixed-coefficient implementation alone.

7 Conclusion

Under the transductive protocol studied here, this paper provides a controlled empirical analysis of sequence pooling for Mamba-based cross-domain ABSA. Mamba Source-only has lower mean Macro-F1 than the reported BERT Source-only reference, while BERT-generated pseudo-label adaptation raises Mamba Last-token Macro-F1 by approximately five points and brings it close to the reported BERT references. Among the tested Mamba pooling rules, Last-token achieves the highest observed mean Macro-F1 in both transfer directions, but the tested scalar-gated Hybrid does not establish Holm-corrected significant superiority over fixed pooling. Minority classes remain weak, calibration metrics worsen after adaptation, and the evaluated Mamba implementation has a larger checkpoint and approximately fourfold higher batch-1 latency than the BERT references on the tested hardware. These findings are limited to the reported backbones, training choices, and transductive protocol and should not be generalized to untested pooling

mechanisms, inductive target generalization, or architecture-isolated efficiency claims.

Statements and Declarations

Funding. The authors did not receive support from any organization for the submitted work. No funding was received to assist with the preparation of this manuscript.

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

Author contributions. Both authors contributed to the study conception and design. Hongzi Xu implemented the models, conducted the experiments, and performed the data analysis. Gaurav Gupta supervised the project and contributed to the methodology and interpretation of results. The first draft of the manuscript was written by Hongzi Xu, and both authors commented on previous versions. Both authors read and approved the final manuscript.

Ethics approval. Not applicable. This study did not involve human participants or animals.

Consent to participate. Not applicable.

Consent for publication. Not applicable.

Data availability. The study uses SemEval-2014 Task 4-style Laptop and Restaurant aspect-level sentiment data. Processed data, experiment configurations, run-level metrics, and the evidence used to reproduce the reported summaries are available from the corresponding author upon reasonable request.

Code availability. The code used to generate the reported results is available from the corresponding author upon reasonable request.

References

- [1] Ben-David S, Blitzer J, Crammer K, et al (2010) A theory of learning from different domains. *Machine Learning* 79:151–175. <https://doi.org/10.1007/s10994-009-5152-4>, URL <https://doi.org/10.1007/s10994-009-5152-4>

- [2] Brier GW (1950) Verification of forecasts expressed in terms of probability. *Monthly Weather Review* 78(1):1–3
- [3] Conneau A, Kiela D, Schwenk H, et al (2017) Supervised learning of universal sentence representations from natural language inference data. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp 670–680, <https://doi.org/10.18653/v1/D17-1070>, URL <https://aclanthology.org/D17-1070/>
- [4] Dao T, Fu DY, Saab KK, et al (2023) Hungry hungry hippos: Towards language modeling with state space models. In: *International Conference on Learning Representations*, <https://doi.org/10.48550/arXiv.2212.14052>, URL <https://arxiv.org/abs/2212.14052>
- [5] Devlin J, Chang MW, Lee K, et al (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pp 4171–4186, <https://doi.org/10.18653/v1/N19-1423>, URL <https://aclanthology.org/N19-1423/>
- [6] Ganin Y, Ustinova E, Ajakan H, et al (2016) Domain-adversarial training of neural networks. *Journal of Machine Learning Research* 17(59):1–35. URL <https://www.jmlr.org/papers/v17/15-239.html>
- [7] Gu A, Dao T (2024) Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752* <https://doi.org/10.48550/arXiv.2312.00752>, URL <https://arxiv.org/abs/2312.00752>
- [8] Gu A, Goel K, Ré C (2022) Efficiently modeling long sequences with structured state spaces. In: *International Conference on Learning Representations*, <https://doi.org/10.48550/arXiv.2111.00396>, URL <https://arxiv.org/abs/2111.00396>
- [9] Guo C, Pleiss G, Sun Y, et al (2017) On calibration of modern neural networks. In: *Proceedings of the 34th International Conference on Machine Learning, Proceedings of Machine Learning Research*, vol 70. PMLR, pp 1321–1330, URL <https://proceedings.mlr.press/v70/guo17a.html>
- [10] Gururangan S, Marasović A, Swayamdipta S, et al (2020) Don’t stop pretraining: Adapt language models to domains and tasks. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp 8342–8360, <https://doi.org/10.18653/v1/2020.acl-main.740>, URL <https://aclanthology.org/2020.acl-main.740/>
- [11] He R, Lee WS, Ng HT, et al (2018) Effective attention modeling for aspect-level sentiment classification. In: *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, pp 1121–1131, URL <https://aclanthology.org/C18-1096/>
- [12] Holm S (1979) A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics* 6(2):65–70. URL <https://www.jstor.org/stable/4615733>
- [13] Loshchilov I, Hutter F (2019) Decoupled weight decay regularization. In: *International Conference on Learning Representations*, URL <https://openreview.net/forum?id=Bkg6RiCqY7>
- [14] Pontiki M, Galanis D, Pavlopoulos J, et al (2014) SemEval-2014 task 4: Aspect based sentiment analysis. In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Association for Computational Linguistics, pp 27–35, <https://doi.org/10.3115/v1/S14-2004>, URL <https://aclanthology.org/S14-2004/>
- [15] Reimers N, Gurevych I (2019) Sentence-BERT: Sentence embeddings using siamese BERT-networks. In: *Proceedings of the 2019*

- Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. Association for Computational Linguistics, pp 3982–3992, <https://doi.org/10.18653/v1/D19-1410>, URL <https://aclanthology.org/D19-1410/>
- [16] Rietzler A, Stabinger S, Opitz P, et al (2020) Adapt or get left behind: Domain adaptation through BERT language model finetuning for aspect-target sentiment classification. In: Proceedings of the Twelfth Language Resources and Evaluation Conference. European Language Resources Association, pp 4933–4941, URL <https://aclanthology.org/2020.lrec-1.607/>
- [17] Sohn K, Berthelot D, Carlini N, et al (2020) Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In: Advances in Neural Information Processing Systems, pp 596–608, URL <https://proceedings.neurips.cc/paper/2020/hash/06964dce9addb1c5cb5d6e3d9838f733-Abstract.html>
- [18] Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. *Information Processing and Management* 45(4):427–437. <https://doi.org/10.1016/j.ipm.2009.03.002>
- [19] Sun C, Huang L, Qiu X (2019) Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, pp 380–385, <https://doi.org/10.18653/v1/N19-1035>, URL <https://aclanthology.org/N19-1035/>
- [20] Wang Y, Huang M, Zhu X, et al (2016) Attention-based LSTM for aspect-level sentiment classification. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pp 606–615, <https://doi.org/10.18653/v1/D16-1058>, URL <https://aclanthology.org/D16-1058/>
- [21] Wilcoxon F (1945) Individual comparisons by ranking methods. *Biometrics Bulletin* 1(6):80–83. <https://doi.org/10.2307/3001968>
- [22] Xu H, Liu B, Shu L, et al (2019) BERT post-training for review reading comprehension and aspect-based sentiment analysis. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, pp 2324–2335, <https://doi.org/10.18653/v1/N19-1242>, URL <https://aclanthology.org/N19-1242/>