VARH-AI at SemEval-2026 Task 10: Exploiting Architectural Diversity with Transformer-SSM Ensembles and Confidence-Based Iterative Refinement for Conspiracy Detection

Hritav S Solanki, Shubham Sharma, Manish Prasad
Rakhi Agrawal, Yashvardhan Sharma
Birla Institute of Technology and Science, Pilani, India
{h20191049, p20240036, p20240903}@pilani.bits-pilani.ac.in
{rakhi.agrawal, yash}@pilani.bits-pilani.ac.in

Abstract

This paper describes our system for SemEval-2026 Task 10 (PsyCoMark), focusing on Subtask 2: binary conspiracy classification in Reddit submission statements. We present a heterogeneous ensemble approach that combines Transformer-based models (DeBERTa, RoBERTa) with State-Space Models (Mamba) to leverage architectural diversity for improved generalization. Our key contributions include: (1) Bidirectional Mamba (BiMamba), adapting state-space sequence models for bidirectional document classification; (2) (2) a safety-switched multi-task training setup that uses marker supervision only for gold-annotated samples, preventing noisy pseudo-labeled rows from affecting the span-extraction objective; and (3) Confidence-Based Iterative Refinement, using committee voting for high-quality pseudo-label generation. Our best official submission achieved a weighted F1 score of 0.78 on the Subtask 2 test set, ranking 4th on the public CodaBench leaderboard. We provide detailed ablation studies demonstrating the complementary contributions of each architectural component to inform future research directions.

1 Introduction

The spread of conspiracy theories on social networks has emerged as a significant social challenge, with implications for public health, democratic processes, and social cohesion (Douglas et al., 2019). Conspiracy narratives often employ distinctive rhetorical patterns, such as identifying shadowy actors, hidden actions, victimized groups, and claimed evidence that distinguish them from legitimate skepticism or investigative journalism (van Prooijen and Acker, 2015). SemEval-2026 Task 10: PsyCoMark (Psycholinguistic Conspiracy Markers) (Samory et al., 2026) addresses this challenge through two subtasks: (1) extraction of psycholinguistic markers (Actor, Action, Victim, Evidence, Effect) and (2) binary classification of Reddit submissions as conspiracy-related or not. In this paper, we focus on Subtask 2, developing a system that achieves competitive performance through architectural diversity and careful training strategies. On the public CodaBench Subtask 2 test leaderboard, our official submission ranked 4th with 0.78.

Our approach is motivated by the complementary strengths of attention-based and state-space architectures. While Transformers provide high-resolution, global context modeling through dense attention (Vaswani et al., 2017), their $O(N^2)$ computational complexity often limits their effective context window. Conversely, State-Space Models (SSMs) like Mamba leverage recurrent state propagation to maintain efficient, linear-time dependencies over long sequences (Gu and Dao, 2023). We hypothesize that an ensemble of these architectures improves robustness by combining the dense representational power of Transformers with the long-context efficiency of Mamba. Our main contributions are:

1. We introduce Bidirectional Mamba for document classification, demonstrating that SSMs can effectively complement Transformer-based approaches for this task.

2. We use a safety-switched multi-task training setup, where the marker-extraction loss is applied only to gold-annotated samples and disabled for pseudo-labeled samples.

3. We provide comprehensive ablation studies quantifying the contribution of each component, identifying key challenges for future work.

Our code is publicly available at: https://github.com/shubham-2001/MambaEnsembleSE26

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 1000–1005

2 Background

2.1 Task Description

The PsyCoMark dataset comprises English Reddit submission statements annotated for conspiracy content and psycholinguistic markers. Subtask 2 requires binary classification of each submission as "Yes" (conspiracy-related) or "No" (not conspiracy-related). The dataset includes 3,822 training samples and 938 test samples. The training set consists of 3,153 labelled samples, exhibiting an incidence rate of 43.55% for conspiracy-related content and contains informal social media language including sarcasm, irony, and domain-specific terminology.

2.2 Related Work

Conspiracy Detection. Prior work has employed pretrained text encoders for conspiracy content classification (Pogorelov et al., 2021), graph neural networks for modeling narrative structure (Shahid and Alonso, 2022), and multimodal approaches combining text with social network features (Moffitt et al., 2021). Strong recent document-classification baselines therefore remain largely transformer-centric, with large RoBERTa and DeBERTa variants serving as competitive pretrained encoders (Liu et al., 2019; He et al., 2021b,a). Our system retains these strong Transformer baselines but augments them with a state-space source and marker-aware supervision.

State-Space Models. Mamba (Gu and Dao, 2023) introduced selective state spaces achieving linear-time complexity while maintaining competitive performance with transformers on language modeling. Applications to classification tasks remain limited, motivating our exploration of BiMamba for conspiracy detection.

Ensemble Methods. Heterogeneous ensembles combining diverse architectures have shown benefits in NLP (Ganaie et al., 2022). Our approach specifically targets *architectural diversity* by combining fundamentally different model families rather than variations of the same architecture. In particular, we mix DeBERTa, Twitter-RoBERTa, and BiMamba sources instead of relying on a homogeneous Transformer-only committee.

Category	Count	Agreement
GT=Yes, LLM=Yes	419	31.4%
GT=Yes, LLM=No	915	–
GT=No, LLM=No	1,695	98.8%
GT=No, LLM=Yes	21	–
Total agreement	2,114	69.3%

Table 1: Agreement between Llama-3.1-70B independent analysis and human labels.

2.3 Exploratory Analysis: Chain-of-Thought Distillation and Annotator Agreement

Before committing to the ensemble architecture, we ran a preliminary Chain-of-Thought (CoT) distillation experiment. Using Llama-3.1-70B (with an 8B fallback), we generated 4-step reasoning traces for 3,050 training samples and then fine-tuned Phi-2 (2.7B, LoRA rank 32–64) on these traces. Rather than forcing the LLM to match the gold label, we flagged disagreements as `LABEL_MISMATCH` and retained only agreement cases for downstream distillation.

The outcome was strongly asymmetric: the LLM agreed with 98.8% of "No" labels but only 31.4% of "Yes" labels, yielding 69.3% overall agreement. After filtering to agreement-only samples, only 373 positive examples remained, which was too few to fine-tune Phi-2 without severe overfitting ; the distilled model defaulted toward "No" and did not beat the smaller encoder baselines. We therefore treated this experiment as diagnostic rather than part of the final system. It nevertheless informed two later choices: using a heterogeneous ensemble to smooth individual model biases, and imposing strict pseudo-label thresholds (Section 3.5) to avoid compounding annotator noise with model noise.

3 System Overview

Figure 1 illustrates our system architecture, consisting of four ensemble sources combined through weighted voting with test-time augmentation.

3.1 Bidirectional Mamba (BiMamba)

Standard Mamba processes sequences unidirectionally, which can be limiting for document classification. We therefore run Mamba in both directions:

$$\mathbf{h}^{fwd} = \text{Mamba}(\mathbf{x}_1, \ldots, \mathbf{x}_T) \quad (1)$$

$$\mathbf{h}^{bwd} = \text{Mamba}(\text{rev}(\mathbf{x}, \mathbf{m})) \quad (2)$$

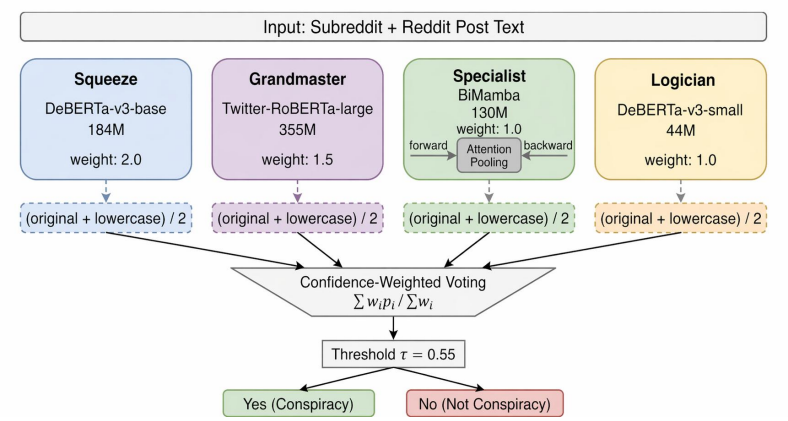

Figure 1: System architecture consisting of four ensemble sources combined through weighted voting with test-time augmentation.

$$\mathbf{h}_i = [\mathbf{h}_i^{fwd} \| \text{rev}(\mathbf{h}^{bwd}, \mathbf{m})_i] \tag{3}$$

where $\text{rev}(\cdot, \mathbf{m})$ is a masked reversal operation that correctly handles variable-length sequences by only reversing valid (non-padding) positions:

$$\text{rev}(\mathbf{x}, \mathbf{m})_i = \mathbf{x}_{L-1-i} \cdot \mathbb{1}[i < L] \tag{4}$$

where $L = \sum_j m_j$ is the valid sequence length. This prevents padding tokens from corrupting the backward representations.

Attention Pooling. Rather than using mean pooling or the final hidden state, we use learned attention pooling:

$$\alpha_i = \frac{\exp(\mathbf{w}^\top \mathbf{h}_i) \cdot m_i}{\sum_j \exp(\mathbf{w}^\top \mathbf{h}_j) \cdot m_j} \quad \mathbf{z} = \sum_i \alpha_i \mathbf{h}_i \tag{5}$$

This allows the model to learn which positions are most informative for classification, providing implicit interpretability.

For our BiMamba implementation, we utilized the `state-spaces/mamba-130m-hf`[1] model (130M params) (Gu and Dao, 2023).

3.2 Transformer Components

We employ three transformer-based models to provide complementary perspectives:

DeBERTa-v3-large ("Squeeze"). Our highest-weight Transformer source uses `microsoft/deberta-v3-large`[2] (304M backbone parameters) and is trained on the synthetically augmented set. We rely on the DeBERTa-v3 architecture (He et al., 2021b,a), whose disentangled attention and improved pre-training objective make it a strong high-capacity classifier for this task.

Twitter-RoBERTa ("Grandmaster"). We include `cardiffnlp/twitter-roberta-large-2022-154m`[3] , a RoBERTa-large checkpoint trained on 154M tweets through December 2022 (Liu et al., 2019; Loureiro et al., 2023). Reddit discourse shares stylistic features with Twitter , including informal language, abbreviations, and hashtags.

DeBERTa-v3-small ("Logician"). Finally, we include `microsoft/deberta-v3-small`[4] (44M backbone parameters) (He et al., 2021b,a). Rather than adding another large model, we use this compact DeBERTa-v3 variant as a low-weight complementary source whose smaller capacity contributes additional regularization and diversity.

3.3 Safety-Switched Multi-Task Learning

To leverage both labeled and pseudo-labeled data effectively, we employ multi-task learning with classification and BIO-tagged span extraction:

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda \cdot v_{ner} \cdot \mathcal{L}_{ner} \tag{6}$$

The key innovation is the safety switch $v_{ner} \in \{0, 1\}$:

$$v_{ner} = \begin{cases} 1 & \text{if sample has gold markers} \\ 0 & \text{if sample is pseudo-labeled} \end{cases} \tag{7}$$

[1] https://huggingface.co/state-spaces/mamba-130m-hf

[2] https://huggingface.co/microsoft/deberta-v3-large

[3] https://huggingface.co/cardiffnlp/twitter-roberta-large-2022-154m

[4] https://huggingface.co/microsoft/deberta-v3-small

Model	Weight	Training Data
Squeeze (DeBERTa-v3-L)	2.0	Augmented
Grandmaster (Tw-RoBERTa)	1.5	Original + Pseudo
Specialist (BiMamba)	1.0	Original
Logician (DeBERTa-v3-S)	1.0	Original

Table 2: Ensemble component weights. Higher weights reflect stronger individual performance on validation.

This prevents the model from learning potentially incorrect marker patterns from pseudo-labeled data while still benefiting from the additional classification signal. The BIO scheme uses 11 tags: O (outside), and B-/I- prefixes for Actor, Action, Victim, Evidence, and Effect.

Architecturally, SS-MTL uses a shared encoder per source model together with task-specific output heads. For Transformer sources, the encoder is the common DeBERTa or RoBERTa backbone; for the Mamba source, it is the bidirectional Mamba trunk. The token-level marker head operates on the full sequence representation, whereas the binary classification head operates on an attention-pooled sequence summary. Thus, the two tasks share the same encoder features but make predictions through independent linear heads.

This explicit shared-encoder classification+BIO formulation corresponds to our universal SS-MTL script. The final submission is trained more heterogeneously: the main stable DeBERTa and BiMamba pipelines are separate classification-oriented scripts and do not force every ensemble source through the same BIO-tagging objective.

3.4 Ensemble Aggregation

We combine predictions through confidence-weighted voting:

$$P(y = 1|\mathbf{x}) = \frac{\sum_i w_i \cdot p_i(\mathbf{x})}{\sum_i w_i} \tag{8}$$

Weights are assigned based on validation performance and diversity considerations (Table 2). Each source contributes 5 fold models, yielding 20 total committee members. The final submission therefore combines exactly four sources: Squeeze (DeBERTa-v3-large), Grandmaster (Twitter-RoBERTa), Specialist (BiMamba), and Logician (DeBERTa-v3-small).

Test-Time Augmentation (TTA). We average predictions on original and lowercased inputs , $\hat{p} = \frac{1}{2}(p(\mathbf{x}) + p(\text{lower}(\mathbf{x})))$, to reduce oversensitivity to erratic casing and rhetorical emphasis in social media text.

3.5 Confidence-Based Iterative Refinement

We augment the labeled training set using a separate high-confidence committee built from the strongest available BiMamba, Twitter-RoBERTa, and DeBERTa fold checkpoints. Let $\bar{p}(\mathbf{x}) = \frac{1}{M}\sum_{i=1}^{M} p_i(\mathbf{x})$ denote the committee-average probability for the conspiracy class. Unlike the final submission ensemble, this pseudo-labeling committee retains only samples far from the decision boundary:

$$L_P(\mathbf{x}) = \begin{cases} \text{Yes} & \text{if } \bar{p}(\mathbf{x}) \geq 0.80 \\ \text{No} & \text{if } \bar{p}(\mathbf{x}) \leq 0.20 \\ \emptyset & \text{otherwise} \end{cases} \tag{9}$$

This conservative filter improves pseudo-label precision at the cost of coverage. From the 938-instance unlabeled official test pool, our 13-checkpoint committee (5 BiMamba, 3 Twitter-RoBERTa, 5 stable DeBERTa) accepts 399 instances, adding 95 pseudo-"Yes" and 304 pseudo-"No" examples. The merged training file therefore contains 3,552 binary-labeled instances with class distribution 1,468 "Yes" and 2,084 "No". Pseudo-labeled rows retain `markers=None`, so SS-MTL disables the marker loss for them.

4 Experimental Setup

4.1 System Configuration

All models were implemented using PyTorch 2.0, Hugging Face Transformers 4.36 and official Mamba implementations. Training was performed on NVIDIA RTX 3090 GPUs with 24GB available video memory. We used AdamW optimization with cosine learning rate scheduling and linear warmup.

4.2 Data Preprocessing

Preprocessing is script-dependent. The BiMamba and stable Transformer pipelines can format inputs as "`Subreddit: {sr}\nText: {text}`" , while the universal SS-MTL script tokenizes the raw `text` field. Otherwise preprocessing is minimal: whitespace normalization, optional URL cleanup, and truncation to 512 tokens.

Synthetic augmentation is instance-level data addition rather than token-level perturbation. We use a fixed auxiliary file

of 1,000 synthetic Reddit-style statements (`train_synthetic_1000.jsonl`), nearly balanced at 507 "Yes" and 493 "No". Concatenating it with the released 3,822 training rows yields `train_augmented_synthetic.jsonl` with 4,822 instances. Because the repository preserves the released files but not the upstream prompts or a standalone filtering script, we describe this component only as a fixed augmentation artifact.

4.3 Training Configuration

Training is heterogeneous across scripts rather than fully uniform. All cross-validation runs use 5 folds, our main pipelines prefer group-based splits keyed by text hashes, and the BiMamba and stable DeBERTa systems tune the classification threshold on each validation fold. For the two main final-ensemble scripts, BiMamba uses learning rate `5e-5`, batch size `8`, gradient accumulation `2`, `10` epochs, dropout `0.20`, weight decay `0.02`, and R-Drop `0.5`, while stable DeBERTa uses learning rate `1e-5`, batch size `4`, gradient accumulation `8`, `6` epochs, dropout `0.10`, weight decay `0.01`, and R-Drop `0.5`. The universal SS-MTL script uses a lighter shared setting (learning rate 2×10^{-5}, batch size 8 , 5 epochs, $\lambda = 1.0$).

Threshold selection is performed strictly on held-out validation predictions during cross-validation. In the BiMamba and stable DeBERTa scripts, we sweep thresholds from 0.05 to 0.95 in 0.01 increments on each validation fold and retain the threshold that maximizes weighted F1 on that fold alone.

Class-weighted cross-entropy addresses class imbalance with $w_c = \frac{N}{2 \cdot N_c}$, where N is the total sample count and N_c is the number of samples in class c.

5 Results

5.1 Official Results

Our final submission achieved a weighted F1 of 0.78 on the official Subtask 2 test set, securing 4th rank on the public CodaBench leaderboard with a score close to the 2nd and 3rd ranked teams, while the top system achieved 0.89.

5.2 Ablation Studies

Table 3 highlights three main findings. First, the strongest standalone sources cluster at 0.73 –0.74 weighted F1. Second, pseudo-labeling adds about 0.01 across architectures, whereas synthetic data yields no further gain. Third, the heterogeneous ensemble improves incrementally from 0.76 (BiMamba + DeBERTa-v3-large) to 0.77 (+ Twitter-RoBERTa) and finally 0.78 after adding Logician (DeBERTa-v3-small). In a controlled same-backbone comparison, SS-MTL improved validation weighted F1 from 0.735 to 0.744, giving an isolated absolute gain of +0.009.

Configuration	W-F1	Δ(%)
Single Models		
BiMamba (Specialist)	0.74	–
DeBERTa-v3-L (Squeeze)	0.73	–
Tw-RoBERTa (Grandmaster)	0.74	–
Training Strategies (vs. Single)		
BiMamba + Pseudo	0.75	+1.35
DeBERTa-v3-L + Pseudo	0.74	+1.37
Tw-RoBERTa + Pseudo	0.75	+1.35
+ *Synthetic (all)*	*no change*	*+0.00*
Ensembles (vs. Best Single: 0.74)		
BiMamba + DeBERTa-v3-L	0.76	+2.70
+ Tw-RoBERTa	0.77	+4.05
+ Logician (DeBERTa-v3-S; Full Ensemble)	**0.78**	**+5.41**

Table 3: 5-Fold cross-validation ablation results. The Δ column indicates relative percentage improvement.

Threshold	W-F1	Performance Indicator
0.45	0.76	● Low confidence
0.50	0.77	● Default threshold
0.55	**0.78**	● **Optimal**
0.60	0.77	● Slightly strict
0.65	0.75	● Too conservative
0.70	0.73	● Overly strict

Symbols: ● *Optimal,* ● *Acceptable,* ● *Suboptimal,* ● *Poor*

Table 4: Effect of classification threshold on weighted F1 score. A threshold of 0.55 yields optimal performance, balancing precision and recall. Color coding and symbols indicate performance quality.

We analyzed the impact of the classification threshold on weighted F1 using cross-validation validation predictions only. The sweep peaked at 0.55 with weighted F1 `0.78`, compared with `0.76` at 0.45, `0.77` at 0.50, `0.77` at 0.60, and `0.73` at 0.70. This indicates that a slightly stricter boundary reduces false positives without discarding too many true positives. For final CodaBench inference, we exported candidate submissions at thresholds 0.45, 0.50, and 0.55.

6 Discussion

BiMamba + DeBERTa-v3-large (0.76) already outperforms either source alone (0.74, 0.73) , supporting the value of architectural diversity. Although BiMamba and Tw-RoBERTa achieve the same standalone weighted F1 of 0.74, our test-

set analysis showed that their predictions were not identical: several samples correctly classified by BiMamba were missed by Tw-RoBERTa, and vice versa. This indicates that the two models make partially complementary errors despite similar aggregate scores. Therefore, BiMamba is retained not as a stronger standalone classifier, but as a non-Transformer source that contributes useful diversity within the ensemble. Pseudo-labeling helps more modestly (+0.01) , indicating that conservative high-confidence additions are useful but still leave some borderline cases unexploited.

7 Conclusion

Our final four-source ensemble of DeBERTa-v3-large, Twitter-RoBERTa, BiMamba, and DeBERTa-v3-small achieved 0.78 weighted F1 on the official test set. The ablations show that the largest gains come from architectural diversity, while pseudo-labeling provides smaller but consistent improvements. The remaining gap likely reflects limited labeled data, ambiguity near the conspiracy-reporting boundary, and residual domain shift. The ensemble requires training and storing multiple medium-to-large models, which increases computational cost; this paper focuses on Subtask 2 and does not present a standalone evaluation for Subtask 1. The system is trained only on English Reddit data, so its behavior on other languages and platforms remains untested. The negligible gain from synthetic augmentation reflects only our fixed LLM-generated dataset; systematic prompt variants and quality-filtering strategies remain important future work.

Acknowledgments

We thank the SemEval-2026 Task 10 organizers for creating the PsyCoMark dataset and evaluation infrastructure. We also thank our professors and mentors at BITS Pilani for providing insightful guidance and computational resources required for the task.

References

Karen M. Douglas, Joseph E. Uscinski, Robbie M. Sutton, Aleksandra Cichocka, Turkay Nefes, Chee Siang Ang, and Farzin Deravi. 2019. Understanding conspiracy theories. *Political Psychology*, 40(S1):3–35.

M. A. Ganaie, Minghui Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan. 2022. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151.

Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*. ArXiv:2312.00752.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *International Conference on Learning Representations*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*. ArXiv:1907.11692.

Daniel Loureiro, Kiamehr Rezaee, Talayeh Riahi, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and Jose Camacho-Collados. 2023. Tweet insights: A visualization platform to extract temporal insights from twitter. *arXiv preprint arXiv:2308.02142*. ArXiv:2308.02142.

Jonathan D. Moffitt, Christopher King, and Kathleen M. Carley. 2021. Hunting for conspiracy theories on social media. *Journal of Information Technology & Politics*, 18(3):304–319.

Konstantin Pogorelov, Daniel Thilo Schroeder, Stefan Brenner, and Johannes Langguth. 2021. WICO graph: A labeled dataset of conspiracy theories and its graph analysis. In *Proceedings of the 2021 Workshop on Open Challenges in Online Social Networks*, pages 29–34. ACM.

Mattia Samory, Felix Soldner, and Veronika Batzdorfer. 2026. SemEval-2026 task 10: PsyCoMark – psycholinguistic conspiracy marker extraction and detection. In *Proceedings of the 20th International Workshop on Semantic Evaluation (SemEval-2026)*.

Fatima Shahid and Omar Alonso. 2022. Detecting and understanding conspiracy theories on social media. In *Proceedings of the ACM Web Conference 2022*, pages 2684–2694. ACM.

Jan-Willem van Prooijen and Michele Acker. 2015. The psychology of conspiracy theories. *Current Directions in Psychological Science*, 24(6):425–430.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.