TeleAI at SemEval-2026 Task 3: Large Language Models for Dimensional Aspect-Based Sentiment Analysis

Yan Zhou[*]**, Shicheng Wang**[*]**, Shiquan Wang, Mengjiao Bao,**
Ruiyu Fang, Shuangyong Song, Yongxiang Li, Xuelong Li[†]
Xingchen AGI Lab, China Telecom Artificial Intelligence Technology (Beijing) Co., Ltd
`2065431274@qq.com, 3269317821@qq.com, wangsq23@chinatelecom.cn`
`baomj@chinatelecom.cn, fangry@chinatelecom.cn, songshy@chinatelecom.cn`
`liyx25@chinatelecom.cn, xuelong_li@chinatelecom.cn`

Abstract

We present TeleAI's system for SemEval-2026 Task 3 (Track A), focusing on Subtask 1 (DimASR): given a text and aspect terms, the system predicts continuous Valence–Arousal (VA) scores under the official $RMSE_{VA}$ metric (Yu et al., 2026). Our regression framework uses Qwen2.5-7B as a feature extractor with LoRA-based parameter-efficient fine-tuning, multilingual/multi-domain mixed training, R-Drop-style consistency regularization, embedding-level PGD adversarial training, output interval mapping, and post-hoc linear calibration. Experimental observations and ablation trends indicate that mixed training and robustness/consistency regularization substantially reduce $RMSE_{VA}$, yielding competitive performance across multiple language–domain settings. We also tried lightweight direct-JSON prompting for Subtasks 2 and 3, but those runs were less competitive, so we focus on Subtask 1.

1 Introduction

Task background and motivation. DimABSA reframes aspect-level sentiment prediction as estimating a point in a two-dimensional affect space: each aspect is assigned Valence (pleasantness) and Arousal (activation/intensity) scores rather than a discrete polarity label. This finer-grained formulation follows affect-space views in emotion psychology and, in SemEval-2026 Task 3 Track A, is evaluated as multilingual/multi-domain VA regression using $RMSE_{VA}$ (Yu et al., 2026).

Overall approach. We model DimASR as end-to-end regression: the input is formed by concatenating `[ASPECT]` and `[TEXT]` segments, and the output is a two-dimensional continuous value (V, A). Our main system uses Qwen2.5-7B as the backbone (obtaining the final-layer hidden states via `AutoModel`), applies mask-aware mean pooling to produce a sequence-level representation, and predicts VA using a 2D linear regression head. To enhance robustness and generalization, we employ LoRA in attention projections, sigmoid-based output interval mapping to $[1.00, 9.00]$, Smooth L1 (Huber) loss, R-Drop-style regression consistency, embedding-level PGD adversarial training, post-hoc linear calibration, and differential learning rates with warmup plus linear decay.

AI Flow framework. This work is conducted within TeleAI's *AI Flow* framework, which advocates device–edge–cloud collaboration and *familial models* for ubiquitous intelligence under resource and communication constraints (An et al., 2026). This context is supported by TeleAI's open model-family reports, including TeleChat, Tele-FLM, TeleChat2/2.5/T1, and TeleChat3-MoE, which document bilingual/multilingual LLM evolution, scaling, post-training, and training infrastructure (He et al., 2024; Wang et al., 2024, 2025d; Liu et al., 2025; Li et al., 2024b,a). Here we focus on AI Flow's NLP modeling component by building an aspect-level VA regression module; our LoRA-based adaptation and exploration of backbone scales align with the *familial model* principle, while explicit multi-tier orchestration is left for future work.

Participation in Subtasks 2 and 3. We also explored Subtasks 2 and 3 with a minimal instruction-prompting setup for direct JSON generation; because it was less competitive under our tuning budget, we focus on Subtask 1.

Key findings (from ablation trends). We observe that **Qwen2.5 models substantially outperform BERT-style encoders**, **multilingual/multi-domain mixed training beats single language–domain training**, **LoRA outperforms full fine-tuning under the same budget**, and **the 7B model**

---

[*] Work done during an internship at TeleAI.

[†] Corresponding author: `xuelong_li@chinatelecom.cn`

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 1846–1852
July 3-4, 2026

outperforms larger models, likely because limited data increases overfitting risk.

Code release. We will release our Subtask 1 training and inference code for reproducibility: `https://github.com/yanzhou06/SemEval-2026-Task3-subtask1-code`

2 Background

2.1 Task definition and I/O format

In Track A Subtask 1 (DimASR), each instance consists of a text and one or more aspect terms appearing in the text. The system must output a pair of real-valued VA scores for each aspect, where each score lies in $[1.00, 9.00]$ and is rounded to two decimals. The official evaluation primarily uses $RMSE_{VA}$ (Yu et al., 2026).

Input example. (synthetic)

```
{
  "ID": "R101",
  "Text": "The screen is crisp and bright, but
  the battery drains fast and feels
  unreliable.",
  "Aspect": ["screen", "battery"]
}
```

Output example. (synthetic)

```
{
  "ID": "R101",
  "Aspect_VA": [
    {"Aspect": "screen", "VA": "7.62#4.85"},
    {"Aspect": "battery", "VA": "2.31#6.40"}
  ]
}
```

Subtasks 2 and 3 (brief note). Subtasks 2 and 3 use a similar JSON-output interface; our direct instruction-following LLM baseline was less competitive, so we mainly report Subtask 1.

2.2 Data split terminology (dev/test)

The shared task provides `dev` and `test` phases (as named by the organizers). In this paper:

dev/test: refer to the official phase splits and evaluation protocol;

val: refers to our locally held-out validation set split from training data for model selection and calibration (different from the official dev set).

2.3 Related work

The valence–arousal (VA) affect space is adopted in this task and its dataset construction as a continuous sentiment representation (Yu et al., 2026; Lee et al., 2026). In NLP, related work includes word-level VAD resources and dimensional sentiment analysis frameworks (Mohammad, 2018), as well as closely related work on microblog hashtag/category classification, e-commerce sentiment analysis and emotional comfort, concept-level emotion-cause detection, emotion-controlled response generation, empathetic dialogue, and text-based emotion detection (Song and Meng, 2015a; Song et al., 2020; Song and Meng, 2015b; Song et al., 2021a,b; Zhao et al., 2023; Wang et al., 2025a,b; Zhao et al., 2026; Wang et al., 2025c). For training methods, we integrate LoRA, R-Drop, and PGD, adapting them to VA regression (Hu et al., 2021; Liang et al., 2021; Madry et al., 2017; Qwen Team, 2025).

Broader perspectives. Our setting can be viewed as a continuous variant of multi-output prediction, which is broadly related to multi-dimensional (multi-target) learning (Jia and Zhang, 2024). To mitigate language/domain shifts, we adopt multilingual and multi-domain mixed training, which connects to general transfer and representation adaptation considerations surveyed in transfer metric learning (Luo et al., 2024). For our lightweight exploration of Subtasks 2–3 via direct JSON generation, we refer to recent surveys on tool learning with language models that summarize structured pipelines and benchmarking practices for LLM-centric task execution (Chen et al., 2025).

3 System Overview

3.1 Pipeline

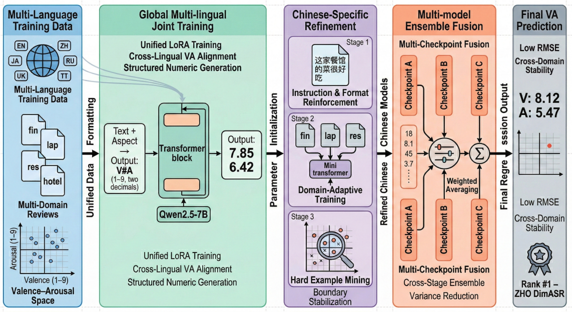

Figure 1: Overview of our DimASR system pipeline: unified LoRA training, Chinese-specific refinement, multi-checkpoint fusion, and final VA prediction.

1. **Parsing and instance expansion:** Expand each JSONL sample into row-wise training instances of the form (Text, Aspect, V, A);
2. **Input templating:** Concatenate inputs using `[ASPECT]` and `[TEXT]` markers;
3. **Qwen2.5-7B + LoRA regression:** Apply mean pooling to obtain sequence representations and regress VA;
4. **Robust training:** Smooth L1 loss with R-Drop consistency and embedding-level PGD adversarial training;
5. **Post-processing and calibration:** Map/clip outputs to $[1, 9]$, fit a linear calibrator on `val`, and apply it to dev/test predictions.

3.2 Data parsing and instance construction

Official training files may contain different structures (e.g., `Aspect_VA`, `Triplet`, `Quadruplet`), so we flatten them into one row per aspect, parse `VA="V#A"` into two labels, and keep the first occurrence for duplicate `(ID, Aspect)` pairs.

3.3 Input representation: explicit aspect and context markers

$$x = \texttt{[ASPECT]}\ a\ \texttt{[/ASPECT] \textbackslash n[TEXT]}\ t\ \texttt{[/TEXT]}$$

where a is the aspect term and t is the text.

3.4 Backbone model and LoRA

We use Qwen2.5-7B final-layer hidden states $H \in \mathbb{R}^{L \times d}$ and attach LoRA adapters to the attention projections (Q/K/V/O and the packed projection), updating only the adapters and regression head (Qwen Team, 2025; Hu et al., 2021). LoRA hyperparameters are $r = 16$, $\alpha = 32$, and dropout $= 0.05$.

3.5 Regression head: mean pooling and interval mapping

Mask-aware mean pooling:

$$h = \frac{\sum_{i=1}^{L} m_i \cdot H_i}{\sum_{i=1}^{L} m_i + \epsilon}, \qquad z = Wh + b \in \mathbb{R}^2$$

Sigmoid-based mapping to $[1.00, 9.00]$:

$$\hat{y} = 1 + 8 \cdot \sigma(z)$$

where $\hat{y} = (\hat{V}, \hat{A})$.

3.6 Training objective: Huber + R-Drop + PGD

We use Smooth L1 (Huber) as the main loss:

$$\mathcal{L}_{\text{main}} = \text{SmoothL1}_{\beta}(\hat{y}, y), \quad \beta = 0.5$$

where $y = (V, A)$ (PyTorch Contributors).

R-Drop-style consistency regularization (regression variant):

$$\mathcal{L}_{\text{rdrop}} = \tfrac{1}{2}\Big(\mathcal{L}_{\text{main}}(\hat{y}^{(1)}, y) + \mathcal{L}_{\text{main}}(\hat{y}^{(2)}, y)\Big) + \alpha \cdot \text{MSE}(\hat{y}^{(1)}, \hat{y}^{(2)}), \quad \alpha = 0.5$$

(Liang et al., 2021)

Embedding-level PGD (ℓ_∞):

$$\delta \leftarrow \Pi_{[-\epsilon, \epsilon]}\big(\delta + \eta \cdot \text{sign}(\nabla_{\delta} \mathcal{L}_{\text{main}}(E + \delta))\big)$$

where $\epsilon = 0.02,\ K = 3,\ \eta = \epsilon / K$ (Madry et al., 2017). The overall loss is:

$$\mathcal{L} = \mathcal{L}_{\text{clean}} + \lambda \cdot \mathcal{L}_{\text{adv}}, \quad \lambda = 0.5$$

3.7 Multilingual and multi-domain mixed training

We scan for `{lang}_{domain}_train_*.jsonl`, split 10% of each file as local `val`, and merge the remaining portions into a multilingual/multi-domain training set. Our best setting uses mixed training for all epochs (`pretrain_ratio=1.0`); multi-stage domain transfer and negative Pearson correlation loss gave no consistent gains and are excluded.

3.8 Post-hoc linear calibration

On `val`, we fit a univariate linear regression calibrator for $\hat{V}$ and $\hat{A}$ separately:

$$\tilde{y} = \text{clip}(a\hat{y} + b, 1, 9)$$

and apply it to dev/test predictions.

4 Experimental Setup

4.1 Data usage and splits

Training set: All `*_train_*.jsonl` files in the training directory (multilingual + multi-domain mixed).

Validation set (val, local split): 10% split from each training file, merged for best checkpoint selection and calibration fitting.

dev/test (official phases): Not used for training; used only to generate submission predictions.

4.2 Hyperparameters and optimization

We use `max_len=128` with right padding. The per-GPU batch size is `per_gpu_bs=4` with `grad_accum=1`; using 8-GPU DDP yields an effective batch size of 32. We optimize with AdamW (weight decay=0.01) and employ differential learning rates: 1e-3 for the regression head and 1e-4 for LoRA parameters, with a warmup ratio of 0.1 followed by linear decay. We apply gradient clipping at 1.0, use bf16 precision, and set `seed=42`. R-Drop uses $\alpha = 0.5$; PGD uses $\epsilon = 2 \times 10^{-2}$, steps=3, and $\lambda = 0.5$.

4.3 Compute resources

Training is conducted on a single machine with 8 NVIDIA A100 (40GB) GPUs using DDP (`torchrun -nproc_per_node=8`). We use bfloat16 and gradient checkpointing to stabilize training.

4.4 Evaluation metrics

Let $\mathbf{y} = (V, A)$. The official metric is:

$$RMSE_{VA} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{y}_p^{(i)} - \mathbf{y}_g^{(i)}\right\|_2^2}.$$

Here N denotes the number of evaluated aspect instances. (Yu et al., 2026). On `val`, we additionally compute PCC_V and PCC_A for analysis, but they are not used for official ranking.

5 Results and Analysis

5.1 Main Results

On the official Track A Subtask 1 evaluation, our proposed system (Qwen2.5-7B with LoRA and robust training strategies) demonstrated highly competitive performance. Specifically, in the English-Restaurant domain, our system achieved a final $RMSE_{VA}$ of 0.85 during the official `dev` phase, ranking first on the leaderboard.

5.2 Ablation Study

To thoroughly evaluate the contribution of each component in our proposed pipeline, we conducted an ablation study on the official `dev` set for the English-Restaurant domain. As shown in Table 1, our baseline model (Qwen2.5-7B using standard full-parameter fine-tuning without specialized techniques) achieved an $RMSE_{VA}$ of 1.03. Our final system significantly reduced the error to 0.85.

Model / Configuration	$RMSE_{VA}$ (↓)
Baseline: Qwen2.5-7B (Full SFT, standard MSE)	1.03
Our Final System (LoRA + All Components)	**0.85**
Replace LoRA with Full SFT	0.91
w/o PGD Adversarial Training	0.94
w/o R-Drop Consistency	0.92
w/o Output Interval Mapping	0.89
w/o Huber Loss (using standard MSE)	0.88
w/o Differential LR & Warmup/Decay	0.87
w/o Post-hoc Linear Calibration	0.87

Table 1: Quantitative ablation study on the `dev` set (English-Restaurant). *w/o* denotes removing a specific component from the final pipeline.

Based on the quantitative results, we summarize the key findings:

Robustness techniques are critical: Removing PGD adversarial training and R-Drop consistency regularization leads to the most significant performance drops (+0.09 and +0.07 $RMSE_{VA}$, respectively), indicating that mitigating overfitting and robustifying the representation space are vital for continuous affect prediction.

LoRA vs. Full Fine-tuning: Under the same training budget, replacing LoRA with full-parameter fine-tuning degrades performance from 0.85 to 0.91. This confirms our hypothesis that parameter-efficient fine-tuning (PEFT) is more suitable for this task, as full fine-tuning on limited domain data tends to overfit.

Model scale and Data composition: Beyond the table, we observed that the Qwen2.5-7B model consistently outperforms its larger variants (specifically, Qwen2.5-14B and Qwen2.5-72B) in our setup. We attribute this to the increased risk of overfitting when fine-tuning massive models on limited domain-specific data. Furthermore, introducing multilingual and multi-domain mixed training provided a universal foundation that stabilized the above components.

Optimization and Calibration: Output interval mapping, Huber loss, differential learning rates, and post-hoc linear calibration collectively contribute to stable convergence and fine-grained error correction, providing a combined gain of approximately 0.05 $RMSE_{VA}$.

Non-effective components: Multi-stage domain transfer training and negative Pearson correlation loss did not improve performance in our preliminary experiments, and thus were excluded from the final configuration.

5.3 Error analysis

We observe the following major error patterns:

(1) Arousal is harder to predict. Arousal depends more on intensity markers, tone, and contrastive structures, and predictions tend to shrink toward mid-range values.

(2) Contrastive structures cause aspect attribution confusion. In sentences like "the food is good, but the service is terrible," negative evidence may incorrectly propagate to unrelated aspects.

(3) Truncation removes crucial evidence. When texts are truncated under `max_len=128`, key trigger words may be dropped; increasing `max_len` to 256 did not significantly improve performance but increased VRAM cost and slowed iteration, so we kept `max_len=128`.

(4) Fine-grained numerical errors. Linear calibration can partially correct global bias, but instance-level fine-grained errors remain.

6 Conclusion

We propose a regression framework centered on Qwen2.5-7B with LoRA, combined with multilingual and multi-domain mixed training, sigmoid-based interval mapping, Huber loss, R-Drop consistency regularization, PGD adversarial training, and post-hoc linear calibration. Our system achieves competitive performance across multiple language–domain settings (Yu et al., 2026). Our analysis suggests that multilingual/multi-domain training and robustness/consistency regularization are the primary sources of improvement, while arousal modeling, contrastive attribution, and truncation remain major error sources.

Limitations

We select models and fit linear calibrators using a locally split validation set (`val`), which may differ in distribution from the official dev/test phases.

Our approach is also limited when the same aspect appears multiple times in one sentence with different evidence: because the input contains only `[ASPECT]` and full `[TEXT]`, the model outputs one VA value and may miss intra-sentential opinion differences. A feasible direction is clause-level segmentation at inference time, possibly with LLM-assisted splitting, to strengthen aspect–opinion alignment; we did not implement it due to time constraints.

Acknowledgments

We thank the organizers of SemEval-2026 Task 3 for the data, evaluation protocol, and shared task platform; the open-source community for PyTorch, Transformers, PEFT, and related tools; and TeleAI's AI Flow initiative, within which this system was developed (An et al., 2026).

References

Hongjun An, Wenhan Hu, Sida Huang, Siqi Huang, Ruanjun Li, Yuanzhi Liang, Jiawei Shao, Yiliang Song, Zihan Wang, Cheng Yuan, Chi Zhang, Hongyuan Zhang, Wenhao Zhuang, and Xuelong Li. 2026. AI flow: perspectives, scenarios, and approaches. *Vicinagearth*, 3:1.

Jinyang Chen, Haolun Wu, Jianhong Pang, Yihua Wang, Dell Zhang, and Changzhi Sun. 2025. Tool learning with language models: a comprehensive survey of methods, pipelines, and benchmarks. *Vicinagearth*, 2:16.

Zhongjiang He, Zihan Wang, Xinzhang Liu, Shixuan Liu, Yitong Yao, Yuyao Huang, Xuelong Li, Yongxiang Li, Zhonghao Che, Zhaoxi Zhang, Yan Wang, Xin Wang, Luwen Pu, Huinan Xu, Ruiyu Fang, Yu Zhao, Jie Zhang, Xiaomeng Huang, Zhilong Lu, and 17 others. 2024. TeleChat technical report. *Preprint*, arXiv:2401.03804.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Bin-Bin Jia and Min-Ling Zhang. 2024. Multi-dimensional classification: paradigm, algorithms and beyond. *Vicinagearth*, 1:3.

Lung-Hao Lee, Liang-Chih Yu, Natalia Loukashevich, Ilseyar Alimova, Alexander Panchenko, Tzu-Mi Lin, Zhe-Yu Xu, Jian-Yu Zhou, Guangmin Zheng, Jin Wang, Sharanya Awasthi, Jonas Becker, Jan Philip Wahle, Terry Ruas, Shamsuddeen Hassan Muhammad, and Saif M. Mohammad. 2026. Dimabsa: Building multilingual and multidomain datasets for dimensional aspect-based sentiment analysis. *Preprint*, arXiv:2601.23022.

Xiang Li, Yiqun Yao, Xin Jiang, Xuezhi Fang, Chao Wang, Xinzhang Liu, Zihan Wang, Yu Zhao, Xin Wang, Yuyao Huang, Shuangyong Song, Yongxiang Li, Zheng Zhang, Bo Zhao, Aixin Sun, Yequan Wang, Zhongjiang He, Zhongyuan Wang, Xuelong Li, and Tiejun Huang. 2024a. 52B to 1T: Lessons learned via Tele-FLM series. *Preprint*, arXiv:2407.02783.

Xiang Li, Yiqun Yao, Xin Jiang, Xuezhi Fang, Chao Wang, Xinzhang Liu, Zihan Wang, Yu Zhao, Xin Wang, Yuyao Huang, Shuangyong Song, Yongxiang Li, Zheng Zhang, Bo Zhao, Aixin Sun, Yequan Wang, Zhongjiang He, Zhongyuan Wang, Xuelong Li, and Tiejun Huang. 2024b. Tele-FLM technical report. *Preprint*, arXiv:2404.16645.

Xiaobo Liang, Lijun Liu, Zhiwei Liu, Yicheng He, and Liang Zhao. 2021. R-drop: Regularized dropout for neural networks. *Preprint*, arXiv:2106.14448.

Xinzhang Liu, Chao Wang, Zhihao Yang, Zhuo Jiang, Xuncheng Zhao, Haoran Wang, Lei Li, Dongdong He, Luobin Liu, Kaizhe Yuan, Han Gao, Zihan Wang, Yitong Yao, Sishi Xiong, Wenmin Deng, Haowei He, Kaidong Yu, Yu Zhao, Ruiyu Fang, and 35 others. 2025. Training report of TeleChat3-MoE. *Preprint*, arXiv:2512.24157.

Yong Luo, Yonggang Wen, Han Hu, Bo Du, Ling-Yu Duan, and Dacheng Tao. 2024. Transfer metric learning: algorithms, applications and outlooks. *Vicinagearth*, 1:2.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *Preprint*, arXiv:1706.06083.

Saif M. Mohammad. 2018. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 english words. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

PyTorch Contributors. Pytorch documentation: `torch.nn.SmoothL1Loss`. `https://pytorch.org/docs/stable/generated/torch.nn.SmoothL1Loss.html`. Accessed 2026-02-23.

Qwen Team. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Shuangyong Song and Yao Meng. 2015a. Classifying and ranking microblogging hashtags with news categories. In *2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS)*, pages 540–541.

Shuangyong Song and Yao Meng. 2015b. Detecting concept-level emotion cause in microblogging. In *Proceedings of the 24th International Conference on World Wide Web Companion*, pages 119–120. Association for Computing Machinery.

Shuangyong Song, Chao Wang, Haiqing Chen, and Huan Chen. 2021a. An emotional comfort framework for improving user satisfaction in E-commerce customer service chatbots. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*, pages 130–137, Online. Association for Computational Linguistics.

Shuangyong Song, Chao Wang, Siyang Liu, Haiqing Chen, Huan Chen, and Hongyun Bao. 2020. Sentiment analysis technologies in AliMe—an intelligent assistant for E-commerce. *International Journal of Asian Language Processing*, 30(04):2050016.

Shuangyong Song, Kexin Wang, Chao Wang, Haiqing Chen, and Huan Chen. 2021b. An emotion-controlled dialog response generation model with dynamic vocabulary. *Preprint*, arXiv:2103.02878.

Shiquan Wang, Ruiyu Fang, Zhongjiang He, Shuangyong Song, and Yongxiang Li. 2025a. Emotional support with LLM-based empathetic dialogue generation. *Preprint*, arXiv:2507.12820.

Shiquan Wang, Ruiyu Fang, Mengxiang Li, Zhongjiang He, and Shuangyong Song. 2025b. Empathetic dialogue generation with LLMs for emotional support. In *Natural Language Processing and Chinese Computing*, volume 16105 of *Lecture Notes in Computer Science*, pages 533–540. Springer.

Shiquan Wang, Mengxiang Li, Shengxiong Peng, Fang Yu, Zhongjiang He, Shuangyong Song, and Yongxiang Li. 2025c. TeleAI at SemEval-2025 task 11: Bridging the gap in text-based emotion detection with prompt engineering and data augmentation. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 102–108, Vienna, Austria. Association for Computational Linguistics.

Zihan Wang, XinZhang Liu, Shixuan Liu, Yitong Yao, Yunyao Huang, Mengxiang Li, Zhongjiang He, Yongxian Li, Luwen Pu, Huinan Xu, Chao Wang, and Shuangyong Song. 2024. TeleChat: An open-source billingual large language model. In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pages 10–20, Bangkok, Thailand. Association for Computational Linguistics.

Zihan Wang, Xinzhang Liu, Yitong Yao, Chao Wang, Yu Zhao, Zhihao Yang, Wenmin Deng, Kaipeng Jia, Jiaxin Peng, Yuyao Huang, Sishi Xiong, Zhuo Jiang, Kaidong Yu, Xiaohui Hu, Fubei Yao, Ruiyu Fang, Zhuoru Jiang, Ruiting Song, Qiyi Xie, and 19 others. 2025d. Technical report of TeleChat2, TeleChat2.5 and T1. *Preprint*, arXiv:2507.18013.

Liang-Chih Yu, Jonas Becker, Shamsuddeen Hassan Muhammad, Idris Abdulmumin, Lung-Hao Lee, Ying-Lung Lin, Jin Wang, Jan Philip Wahle, Terry Ruas, Alexander Panchenko, Ilseyar Alimova, Kai-Wei Chang, Lilian Wanzare, Nelson Odhiambo, Bela Gipp, and Saif M. Mohammad. 2026. SemEval-2026 task 3: Dimensional aspect-based sentiment analysis (DimABSA). In *Proceedings of the 20th International Workshop on Semantic Evaluation (SemEval-2026)*. Association for Computational Linguistics.

Deji Zhao, Donghong Han, Ye Yuan, Bo Ning, Yongxiang Li, Zhongjiang He, Chao Wang, and Shuangyong Song. 2026. MEGE: A mixed emotion graph model for empathetic dialogue generation. *Neural Networks*, 195:108252.

Deji Zhao, Donghong Han, Ye Yuan, Chao Wang, and Shuangyong Song. 2023. MuSE: A multi-scale emotional flow graph model for empathetic dialogue generation. In *Machine Learning and Knowledge Discovery in Databases: Research Track*, volume 14170 of *Lecture Notes in Computer Science*, pages 491–507. Springer.

Hyperparameter	**Value / Setting**
Epochs	10 (`pretrain_ratio=1.0` at best config)
Max Sequence Length	128
Batch Size	per_gpu=4, grad_accum=1
Optimizer	AdamW, $\lambda_{\text{wd}} = 0.01$
Learning Rate	$\eta_{\text{head}} = 10^{-3}$; $\eta_{\text{LoRA}} = 10^{-4}$; warmup 0.1 + linear decay
Precision	bf16 (grad clip=1.0, `seed=42`)
LoRA Config	r=16, α=32, p_{drop}=0.05
LoRA Targets	q_proj,k_proj,v_proj,o_proj,W_pack
Loss Function	SmoothL1 (β=0.5)
Regularization	R-Drop (α=0.5), PGD (ϵ=2e-2, steps=3, λ=0.5)
Calibration	Univariate linear regression on V and A, clip to [1,9]

Table 2: Hyperparameter settings for our best model.

A Appendix: Hyperparameters and Pseudocode

A.1 Key hyperparameters

A.2 Training pseudocode (simplified)

Algorithm 1 Simplified Training Procedure with R-Drop and PGD

```
Require: Epochs $E$, loader, Model, $\alpha$, $\lambda$, PGD steps $K$, bound $\epsilon$
for epoch $= 1$ $E$ do
    for batch $(x, y)$ loader do
                                        ▷ 1. R-Drop Forward Pass
        $y_1 \leftarrow \text{model}(x)$
        $\mathcal{L}_1 \leftarrow \text{SmoothL1}(y_1, y)$
        $y_2 \leftarrow \text{model}(x)$
        $\mathcal{L}_2 \leftarrow \text{SmoothL1}(y_2, y)$
        $\mathcal{L}_{\text{mse}} \leftarrow \text{MSE}(y_1, y_2)$
        $\mathcal{L}_{\text{clean}} \leftarrow 0.5 \times (\mathcal{L}_1 + \mathcal{L}_2) + \alpha \times \mathcal{L}_{\text{mse}}$
                                        ▷ 2. PGD Adversarial Steps
        $\delta \leftarrow 0$
        for $k = 1$ $K$ do
            $\mathcal{L}_{\text{step}} \leftarrow \text{SmoothL1}(\text{model}(x, E + \delta), y)$
            $g \leftarrow \text{sign}(\nabla_\delta \mathcal{L}_{\text{step}})$
            $\delta \leftarrow \text{clip}(\delta + \text{step} \times g, [-\epsilon, \epsilon])$
        end for
        $\mathcal{L}_{\text{adv}} \leftarrow \text{SmoothL1}(\text{model}(x, E + \delta), y)$
                                        ▷ 3. Final Loss and Optimization
        $\mathcal{L} \leftarrow \mathcal{L}_{\text{clean}} + \lambda \times \mathcal{L}_{\text{adv}}$
        backward($\mathcal{L}$)
        optimizer.step()
    end for
end for
```