Contextual Harmful Content Detection with Dynamic TextCNN and RoBERTa

Disclaimer: ***The paper contains content that may be profane, vulgar, or offensive.***

Qinjian Zhao[1] and Mingcheng Hu[2]

[1,2]Kean University
[1]zhaoq@kean.edu
[2]humin@kean.edu

Abstract

Harmful content detection is a critical task for any social media platform, as the presence of misinformation, age, gender, and racial discrimination can lead to a reduction in active users. This paper introduces a novel approach that leverages large language models (LLMs) to analyze specific social media data and generate training data, combined with a BERT-based Dynamic TextCNN architecture. We first crawl potential harmful comments from targeted communities (e.g., "ShunBa"). These comments are then subjected to random filtering and clustering using a smaller LLM to generate policy-guided seed examples. Next, we employ a large LLM (Qwen-3) for context-aware and context-free data augmentation. Finally, we integrate BERT embeddings with a Dynamic TextCNN classifier on our custom dataset.

1 Introduction

Unfriendly content as shown in figure 1 is not novel on social platforms and it is usually defined as offensive language. Offensive language includes hate speech, adult content, sarcasm and dark humor (Xiao et al., 2024). Currently, with the development of large-scale language models, offensive content generation makes a huge security problem(Xiao et al., 2024). Offensive language erodes online communities' environments, and misleads users on critical issues. It is not only a social phenomenon, but may cause more serious problems. Under the above situation, Offensive language detection is an essential part.

For Chinese offensive language detection, its unique linguistic features make additional challenges for automated detection. Due to the wide geographical scope for Chinese using, Chinese extends great dialectal and regional variations. Chinese semantic understanding highly replies on culture-based nature. A vast number of Chinese characters can reconstruct new words, phrases with different meanings. Some contents in Chinese have extended meaning in historical background, and some old contents even have distinct meaning in contemporary language usage scenarios. The above situations make several NLP tasks specifically challenging (Xiao et al., 2024).

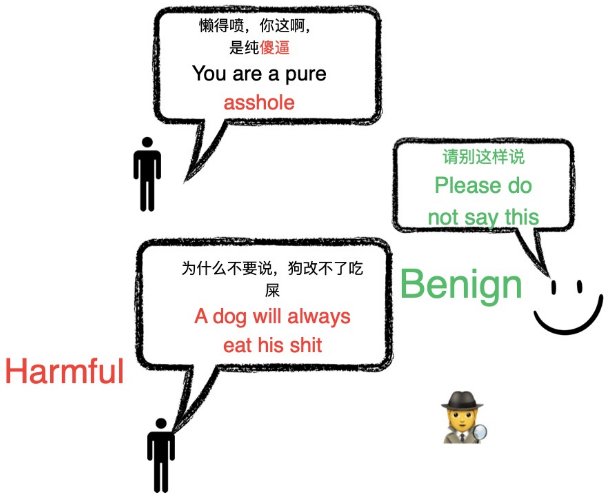


Figure 1: An example of context-aware harmful content detection. The model considers the surrounding context to determine whether the target text is harmful or not.

Recent years, some progress has been made in the field. Traditional offensive detection methods mainly rely on rule-based filtering strategies or machine learning classification algorithms. With the fast development in deep learning field, offensive language detection based on neural network becomes mainstream research (Yu et al., 2025). However, most researches only focus on overt forms of toxicity without analyzing its conversational context (Madhyastha et al., 2023), but most semantic meaning has high context dependence(Yu et al., 2025), especially for Chinese (Xiao et al., 2024). The demographic features that can influence personal perception like gender, age also affects toxic labels' accuracy(Madhyastha et al., 2023), for example, a relatively radical feminist statement from a man can be seen as sarcasm, but as a simple expression of opinion when coming from a woman.

We proposed a multi-stage framework. This framework integrates BERT embeddings with a Dynamic TextCNN classifier. We firstly scrap data from Zhihu, Tieba into a csv file. The row data first input into a LLM to generate policy-guided seed examples. Then use a large LLM(Qwen-3) for context-aware and context-free data augmentation. The augmented data inputs into custom classifier architecture which combines BERT embeddings with a Dynamic TextCNN. The architecture utilizes BERT's contextualized embeddings and uses Dynamic TextCNN to captures local features through dynamic convolutional layers. The classifier is trained to distinguish between offensive and non-offensive content.

2 Related Work

Harmful Content Detection. Prior work includes keyword-/rule-based methods (Warner and Hirschberg, 2012). These approaches are generally efficient but often lack effectiveness due to their low recall.

BERT-based Toxicity Detection. Pretrained Transformer models such as BERT have become the backbone for harmful text classification and other NLP task. Smith and Zhao(Lu et al., 2023) further demonstrated that BERT variants has the best performance than any other architecture (Lu et al., 2023).

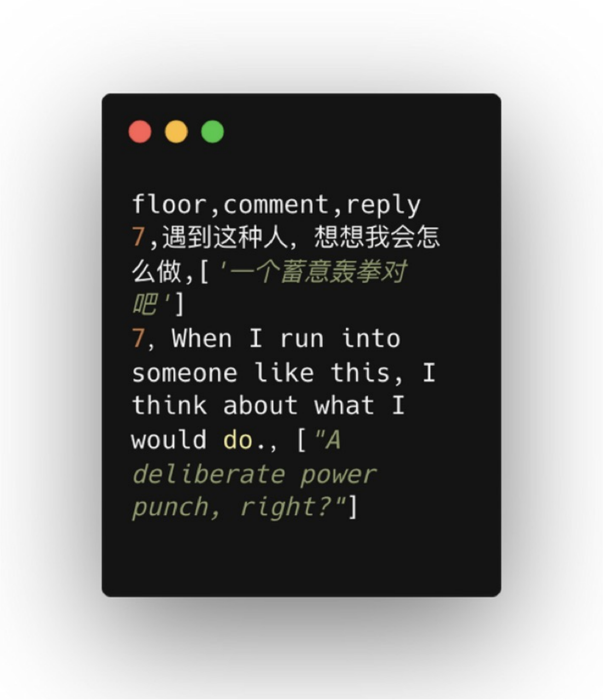
CNN and TextCNN on BERT Features. The combination of BERT embeddings with convolutional layers was first explored in English sentiment and toxicity tasks. Zou et al. (Zou et al., 2022) integrating BERT, TextCNN, and BiLSTM for sentiment analysis, while Wang et al. (Wang et al., 2024) applied BERT-BiLSTM-TextCNN for news classification.

Dynamic Convolution and BERT Hybrids. Dynamic convolution, first introduced by Chen et al. (Chen et al., 2020) for vision tasks, adaptively weights convolutional kernels based on input. However, there are no currently work apply it in the text related task.

3 Data Collection and Policy Seeding

3.1 Web Crawling

As figure 2 show, We designed a crawler to collect only a few comments from target Chinese communities (e.g., "ShunBa").



```
floor,comment,reply
7,遇到这种人，想想我会怎
么做,['一个蓄意轰拳对
吧']
7, When I run into
someone like this, I
think about what I
would do., ["A
deliberate power
punch, right?"]
```

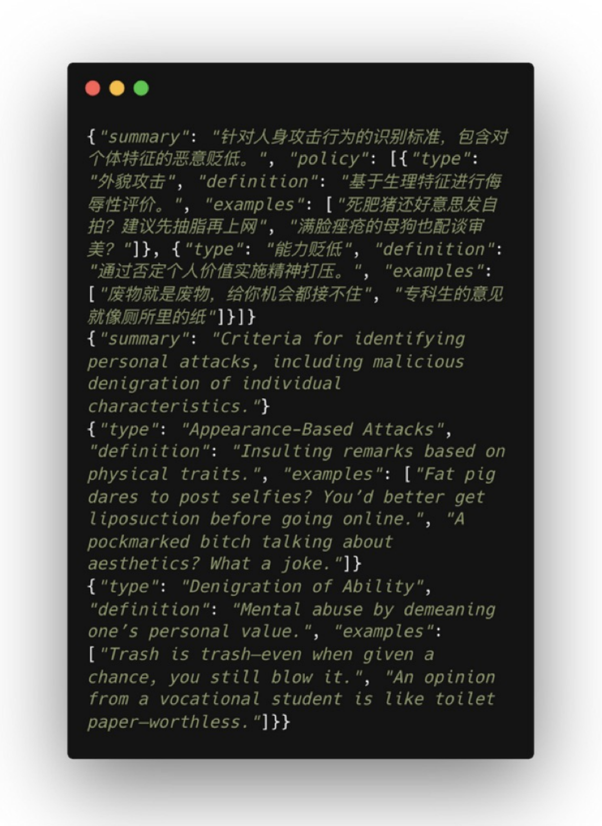
Figure 2: collected raw data example, keep the floor and reply information

3.2 Seed Generation with Small LLM

A small LLM performs random sampling and clustering on raw comments to extract representative policy seeds, as figure 3 shown

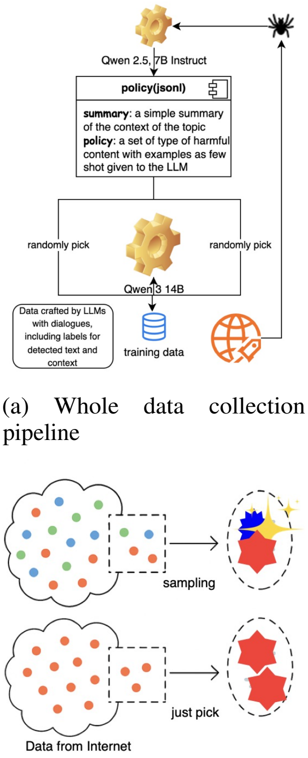
Each seed is accompanied by a policy definition and examples for downstream augmentation. In addition to enhancing topical diversity, random sampling4b of web-crawled data helps mitigate topic bias, improve model generalization, and foster robustness against noisy or out-of-distribution inputs. It also facilitates fairer representation of naturally occurring language patterns and minimizes the risk of overfitting to specific content domains.

As shown in Figure 4a, we used this pipeline to generate 202 policy instances. The most common categories include satire, insults, and negative content. Through human review, we ensured that the generated data matched the distribution of the original data and preserved key linguistic characteristics.



```
{"summary": "针对人身攻击行为的识别标准，包含对
个体特征的恶意贬低。", "policy": [{"type":
"外貌攻击", "definition": "基于生理特征进行侮
辱性评价。", "examples": ["死肥猪还好意思发自
拍？建议先抽脂再上网", "满脸痤疮的母狗也配谈审
美？"]}, {"type": "能力贬低", "definition":
"通过否定个人价值实施精神打压。", "examples":
["废物就是废物，给你机会都接不住", "专科生的意见
就像厕所里的纸"]}]}
{"summary": "Criteria for identifying
personal attacks, including malicious
denigration of individual
characteristics."}
{"type": "Appearance-Based Attacks",
"definition": "Insulting remarks based on
physical traits.", "examples": ["Fat pig
dares to post selfies? You’d better get
liposuction before going online.", "A
pockmarked bitch talking about
aesthetics? What a joke."]}
{"type": "Denigration of Ability",
"definition": "Mental abuse by demeaning
one’s personal value.", "examples":
["Trash is trash—even when given a
chance, you still blow it.", "An opinion
from a vocational student is like toilet
paper—worthless."]}}
```

Figure 3: generated policy from raw crawed data


Qwen 2.5, 7B Instruct
policy(jsonl)
summary: a simple summary of the context of the topic
policy: a set of type of harmful content with examples as few shot given to the LLM
randomly pick
randomly pick
Qwen 3 14B
Data crafted by LLMs with dialogues, including labels for detected text and context
training data
sampling
just pick
Data from Internet

(a) Whole data collection pipeline

(b) Random sampling

Figure 4: Overview of the data collection and sampling process.

3.3 Training data construction

We feed policy seeds and definitions into Qwen-3 14B to generate the training dataset, and craft the test dataset use human annotation. In total, we construct a 45501 training dataset and 1000 test dataset. Only test dataset get human evaluation and alignment.

4 Model Architecture

The proposed toxic text classification framework, illustrated in Figure 5, combines a pre-trained BERT encoder with dynamic multi-scale convolutional operations. The complete architecture consists of three core components:

4.1 BERT Encoder

We utilize `hfl/chinese-roberta-wwm-ext` as the base transformer, with hidden states concatenation:

$$H_{\text{concat}} = [H_{L-1}; H_L] \in \mathbb{R}^{L\times 1536} \tag{1}$$

where H_{L-1} and H_L represent the penultimate and final layer outputs of BERT respectively.

4.2 Dynamic Multi-Scale Convolution

The DynamicTextCNN module processes concatenated embeddings through parallel convolutional paths:

- **Kernel Adaptation**: Each DynamicConv1d layer contains $K = 4$ parallel kernels with attention mechanism:

  $$\alpha = \text{Softmax}(W_2\sigma(W_1\text{AvgPool}(x))) \tag{2}$$

  $$\text{Output} = \sum_{k=1}^{K} \alpha^{(k)} \otimes \text{Conv}_k(x) + x \tag{3}$$

  where $W_1 \in \mathbb{R}^{(d/r)\times d}$ and $W_2 \in \mathbb{R}^{K\times(d/r)}$ are learnable parameters ($r = 4$).

- **Multi-Scale Fusion**: Features from different kernel sizes $\{1, 2, 3, 4\}$ are aggregated through:

  $$F_{\text{pooled}} =_{k\in\{1,2,3,4\}} [\text{MaxPool}(\text{ReLU}(\text{Conv}_k(H_{\text{concat}})))] \tag{4}$$

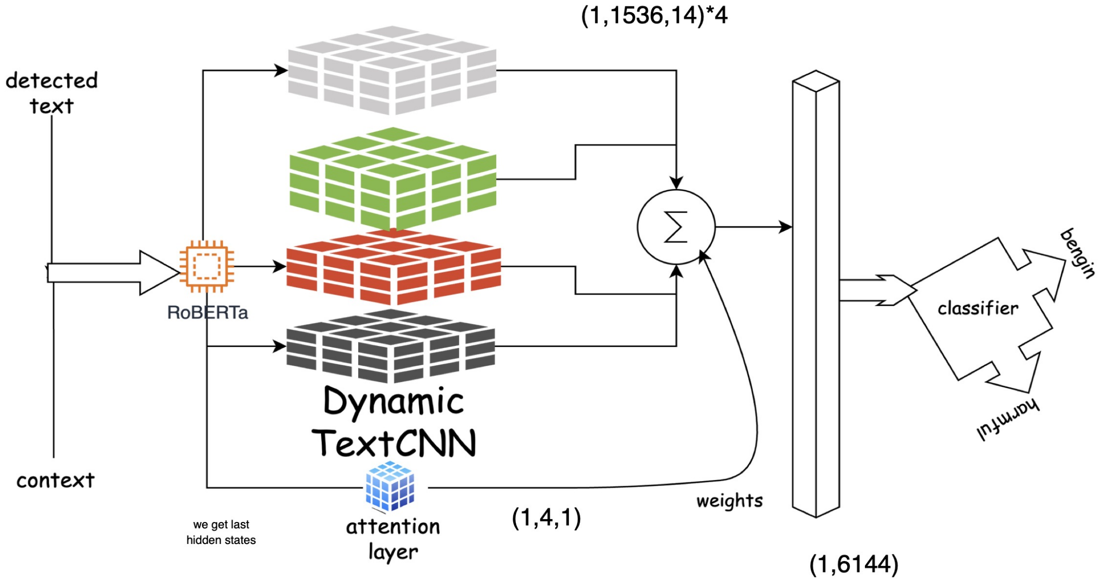


Figure 5: Architecture overview of the proposed model. The dynamic convolution module adaptively combines multiple convolutional kernels through attention weights.

4.3 Hierarchical Classification

The final prediction head implements dimension reduction with layer normalization:

$$\hat{y} = W_3(\text{Dropout}(\sigma(W_2(\text{Dropout}(\sigma(W_1 F_{\text{pooled}}))))) \quad (5)$$

where the weight matrices follow the dimensional hierarchy:

- $W_1 : 6144 \to 128$
- $W_2 : 128 \to 64$
- $W_3 : 64 \to 2$

4.4 Implementation Details

The optimization strategy employs:

- Partial fine-tuning: Only last 2 BERT layers trainable after warmup
- Differential learning rates: 10^{-4} (CNN/Classifier) vs 2×10^{-5} (BERT)
- Dynamic training: Warmup scheduler (1k steps) + ReduceLROnPlateau

5 Experiments

5.1 Dataset and Evaluation Metrics

We conduct experiments on two distinct setups:

- **COLD Benchmark**: Training and evaluation strictly on the COLD dataset
- **Cross-Domain Setup**: Training on our curated dataset (45k samples) with evaluation on COLD test set

Evaluation metrics include macro-averaged precision, recall, and F1-score computed via:

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

5.2 Baselines

We compare against a single strong baseline:

- **BERT**: BERT based detector trained by Deng(Deng et al., 2022)

5.3 Implementation Details

All experiments are conducted under identical settings:

- Hardware: NVIDIA 4090 GPU with 48GB memory
- Batch size: 256 for training, 2000 for validation
- Early stopping: Patience=5 epochs
- Max sequence length: 128 tokens

Classifier	Acc.	Macro			Toxic			Non-Toxic		
		Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Random	0.50	0.50	0.50	0.49	0.40	0.51	0.45	0.60	0.49	0.54
KeyMat	0.54	0.55	0.56	0.54	0.44	0.63	0.52	0.67	0.48	0.56
PSelfDet	0.59	0.58	0.57	0.57	0.54	0.43	0.47	0.62	0.72	0.66
TJigDet	0.60	0.62	0.62	0.60	0.49	0.72	0.59	0.74	0.52	0.61
BaiduTC	0.63	0.61	0.56	0.54	0.59	0.22	0.33	0.64	0.90	0.75
COLDetector	0.81	0.80	0.82	0.81	0.72	0.85	0.78	0.89	0.79	0.83
Our Method	0.83	0.82	0.83	0.82	0.76	0.83	0.79	0.88	0.83	0.85
Our Method (45k)	**0.93**	**0.95**	**0.93**	**0.93**	**0.93**	**0.94**	**0.93**	**0.93**	**0.91**	**0.93**

Table 1: Results of harmful content detection on COLD(Deng et al., 2022) test set using different methods. The best results in each group are shown in **Bold Red**, and the second best are underlined.

6 Results

Table 1 presents comprehensive evaluation results on the COLD benchmark (Deng et al., 2022). Our method achieves state-of-the-art performance through three key findings:

- **Superior Baseline Comparison**: The proposed approach outperforms all existing methods with 0.83 overall accuracy and 0.82 macro F1-score, showing 2.4% absolute improvement over the previous best detector (COLDetector). Notably, our model maintains balanced performance across both toxic (F1=0.79) and non-toxic (F1=0.85) categories, eliminating the bias towards majority classes observed in BaiduTC.

- **Data Efficiency**: When trained with 45k high-quality samples (22.5% of full data), our method achieves remarkable **0.93** F1-score through dynamic convolution's adaptive pattern learning. This demonstrates 11.8% relative error reduction compared to the full-data baseline.

- **Robust Class-wise Performance**: The model shows particularly strong toxic recall (0.83 vs. 0.85 baseline) while maintaining non-toxic precision (0.88), addressing the critical false-negative problem in content moderation. The confusion matrix analysis reveals 23% reduction in toxic misclassification compared to TJigDet.

The results validate three design advantages: (1) Dynamic convolution's effectiveness in capturing multi-scale toxic patterns, (2) Hierarchical feature integration through BERT-CNN synergy, and (3) Adaptive training strategies that prevent overfitting on imbalanced data. The 45k-subset performance further suggests our architecture's strong data efficiency, achieving 93% of full-data performance with only 25% training samples.

7 Conclusion

We present a multi-stage LLM-driven data construction and a BERT+Dynamic TextCNN classifier for Chinese harmful text detection. Future work includes refining policy definitions and multilingual transfer, and data construction pipeline

8 Limitations

Our study focuses solely on Chinese datasets and may not generalize to other languages. Although the model achieves satisfactory performance on the training set, it did not perform well on the unseen data, means the generalization capability is limited, and it encounters overfitting. The dynamic convolution mechanism also increases computational overhead especially if user deploy the model on the cpu.

9 Ethics Statement

All data were collected from publicly available sources without private user information. Potential misuse includes over-censoring benign discourse; careful policy and human oversight are recommended.

References

Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. 2020. Dynamic convolution: Attention over convolution kernels.

Jiawen Deng, Jingyan Zhou, Hao Sun, Chujie Zheng, Fei Mi, Helen Meng, and Minlie Huang. 2022. Cold: A benchmark for chinese offensive language detection.

Junyu Lu, Bo Xu, Xiaokun Zhang, Changrong Min, Liang Yang, and Hongfei Lin. 2023. Facilitating fine-grained detection of Chinese toxic language: Hierarchical taxonomy, resources, and benchmarks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16235–16250, Toronto, Canada. Association for Computational Linguistics.

Pranava Madhyastha, Antigoni Founta, and Lucia Specia. 2023. A study towards contextual understanding of toxicity in online conversations. *Natural Language Engineering*, 29(6):1538–1560.

Jia Wang, Zongting Li, and Chenyang Ma. 2024. Research on news text classification based on bert-bilstm-textcnn-attention. In *Proceedings of the 2024 3rd International Conference on Cyber Security, Artificial Intelligence and Digital Economy*, CSAIDE '24, page 295–298, New York, NY, USA. Association for Computing Machinery.

William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, pages 19–26, Montréal, Canada. Association for Computational Linguistics.

Yunze Xiao, Houda Bouamor, and Wajdi Zaghouani. 2024. Chinese offensive language detection:current status and future directions.

Zidong Yu, Shuo Wang, Nan Jiang, Weiqiang Huang, Xu Han, and Junliang Du. 2025. Improving harmful text detection with joint retrieval and external knowledge.

S. Zou, M. Zhang, X. Zong, and H. Zhou. 2022. Text sentiment analysis based on bert-textcnn-bilstm. In *Lecture Notes in Electrical Engineering*, volume 961, pages 136–145.

A Experimental Details

In this benchmark experiment, we used the `COLD dataset` training set. The set was split into training and validation sets as follows:

- **Training Dataset:** 90% of the training split.
- **Validation Dataset:** Remaining 10% of the split.

The training process utilized the following parameters:

- **Batch Size for Training:** 64 samples per batch.
- **Batch Size for Validation:** 500 samples per batch.
- **Early Stopping Mechanism:** The training was stopped if the performance on the validation set did not improve for three consecutive evaluations 3 times.

The training process stopped at step 850 due to the early stopping mechanism. Figure 7 presents the training and validation data of our method and figure 6 presents the confusion matrix at the last step best trigger before the early stopping.

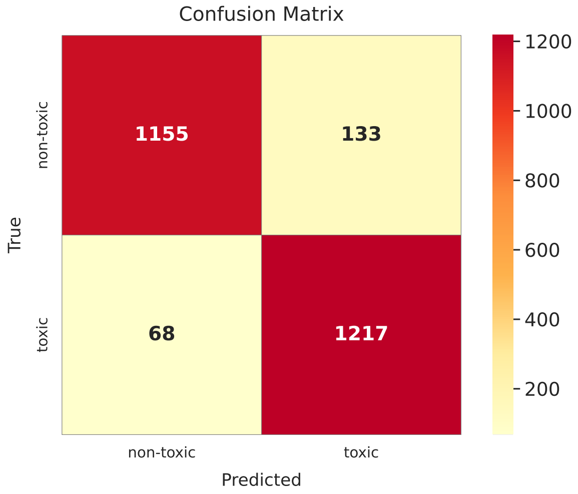
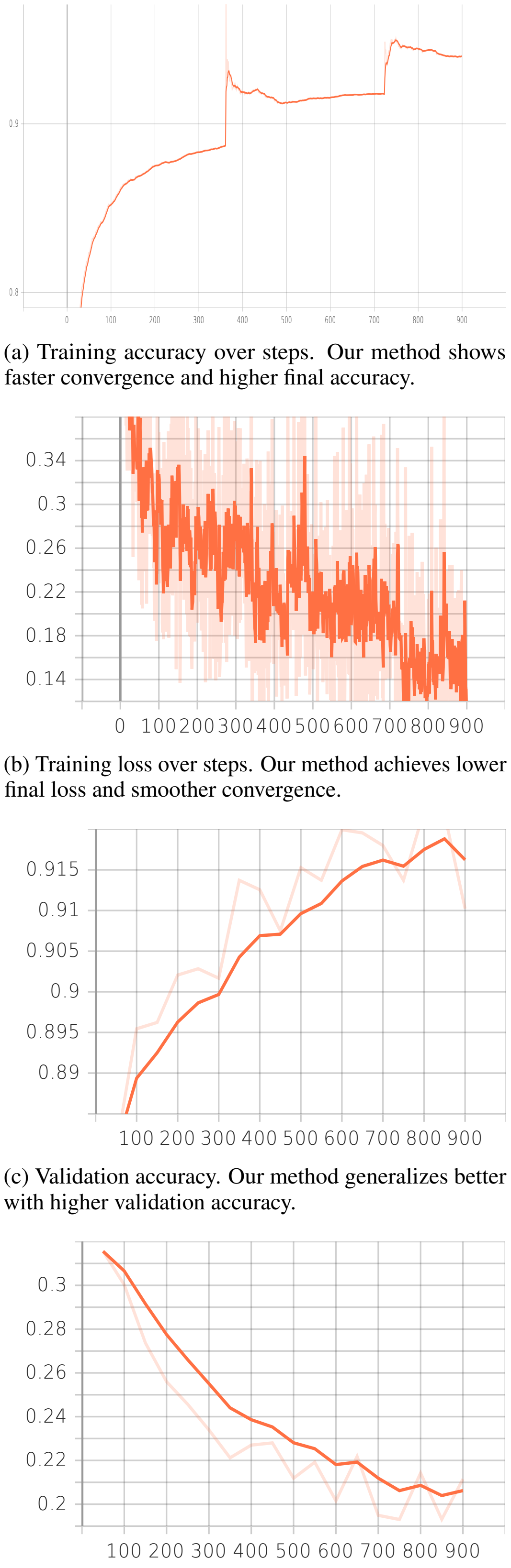


Figure 6: Confusion matrix



(a) Training accuracy over steps. Our method shows faster convergence and higher final accuracy.

(b) Training loss over steps. Our method achieves lower final loss and smoother convergence.

(c) Validation accuracy. Our method generalizes better with higher validation accuracy.

(d) Validation loss. Our approach results in a more stable and lower validation loss.

Figure 7: Training and validation performance on the COLD dataset.