ELMO EMBEDDING

ELMO - TRAINABLE LAMBDA

Parameter setting

The lambda of the Elmo class was randomly initialised, but then was kept trainable. So, when the LSTM weights were frozen.

TEST - DATA CLASSIFICATION REPORT

	precision	recall	f1-score	support
0	0.92	0.91	0.92	1900
1	0.96	0.97	0.97	1900
2	0.89	0.85	0.87	1900
3	0.86	0.90	0.88	1900
accuracy			0.91	7600
macro avg	0.91	0.91	0.91	7600
weighted avg	0.91	0.91	0.91	7600

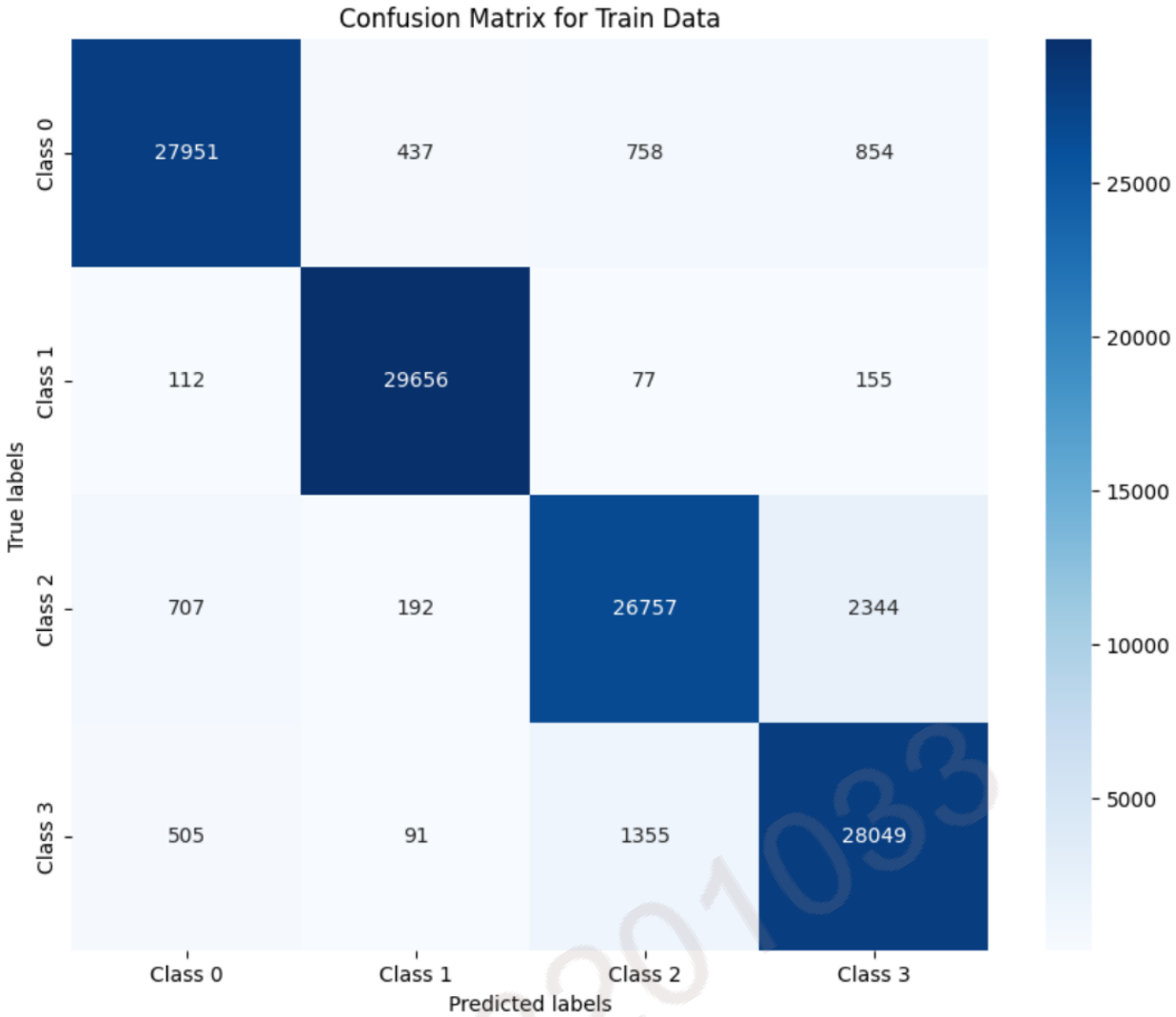
- The overall accuracy of the model on the test set is 0.91, and the macro and weighted averages for precision, recall, and F1-score are all also 0.91,
- shows a balanced performance across all classes without significant bias towards any particular class.

TRAIN - DATA CLASSIFICATION REPORT

	precision	recall	f1-score	support
0	0.95	0.93	0.94	30000
1	0.98	0.99	0.98	30000
2	0.92	0.89	0.91	30000
3	0.89	0.93	0.91	30000
accuracy			0.94	120000
macro avg	0.94	0.94	0.94	120000
weighted avg	0.94	0.94	0.94	120000

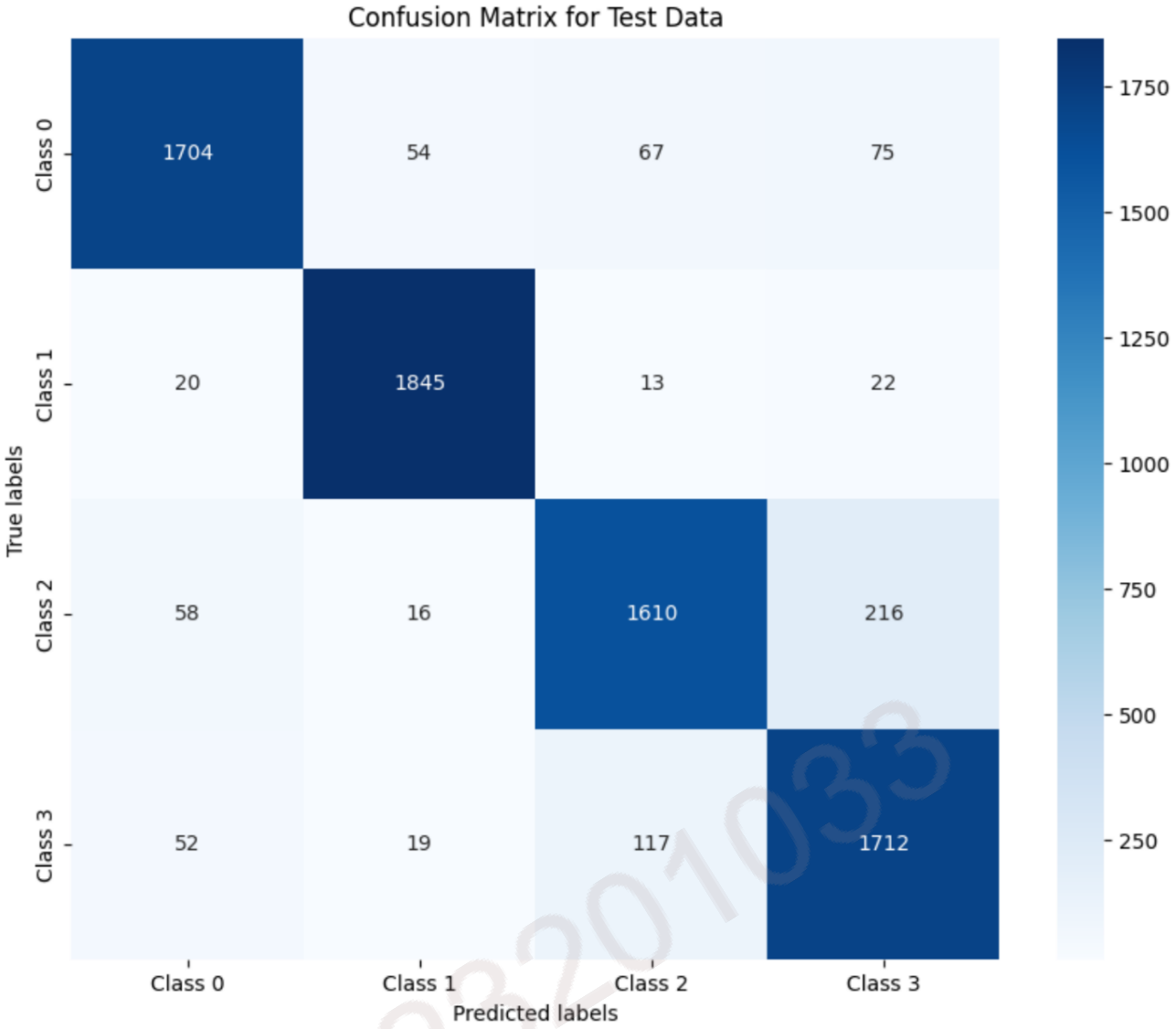
- classification report for the training data shows that the model has achieved a high level of precision, recall, and F1-score across all four classes, indicating a strong performance on the training set.
- Class 1 has the highest precision and recall, yielding the highest F1-score of 0.98, while class 3 has the lowest precision but still a high recall, resulting in the lowest F1-score of 0.91.
- Overall, the model's accuracy on the training set is 0.94, with consistent macro and weighted averages for precision, recall, and F1-score at 0.94.

CONFUSION MATRIX TRAINING DATA



- Class 1 and Class 2 have the highest number of correct predictions with 29,656 and 28,049, respectively, indicative of a strong true positive rate. Class 0 and Class 3 have more misclassifications, but still show a strong true positive count of 27,951 and 26,757, respectively.
- Misclassifications for each class primarily occur with adjacent classes (Class 0 with Class 1, Class 2 with Class 3), which could indicate some feature overlap or ambiguity between these classes.
- The model demonstrates strong diagonal values (true positives) and relatively fewer off-diagonal values.

CONFUSION MATRIX TEST DATA



- The majority of predictions concentrated along the matrix's diagonal, indicating correct classifications.
- Class 1 has the highest number of correct predictions (1845), suggesting that the model is most effective at identifying this class.
- Some confusion is observed between Class 2 and Class 3, as evidenced by the relatively higher number of misclassifications between these two classes.
- Model's ability to generalize from the training to the test data remains strong.

ELMO - FROZEN LAMBDA

Parameter setting

The lambda of the Elmo class was randomly initialised and it was frozen. So, when the classifier was being trained , the whole ELMO was only on eval mode as there is nothing to be

changed here.

TEST - DATA CLASSIFICATION REPORT

	precision	recall	f1-score	support
0	0.81	0.87	0.84	1900
1	0.97	0.84	0.90	1900
2	0.85	0.79	0.82	1900
3	0.79	0.88	0.83	1900
accuracy			0.85	7600
macro avg	0.85	0.85	0.85	7600
weighted avg	0.85	0.85	0.85	7600

- This classification report for the test data indicates that Class 1 has the highest precision at 0.97, indicating a low number of false positives, and an impressive F1-score of 0.90, denoting a balanced precision-recall relationship for this class.
- Class 3 has the lowest precision but the highest recall, suggesting it has more false positives but fewer false negatives.
- The overall accuracy of the model is 0.85, and the macro and weighted averages for precision, recall, and F1-score are also 0.85, showing consistent performance across all classes.

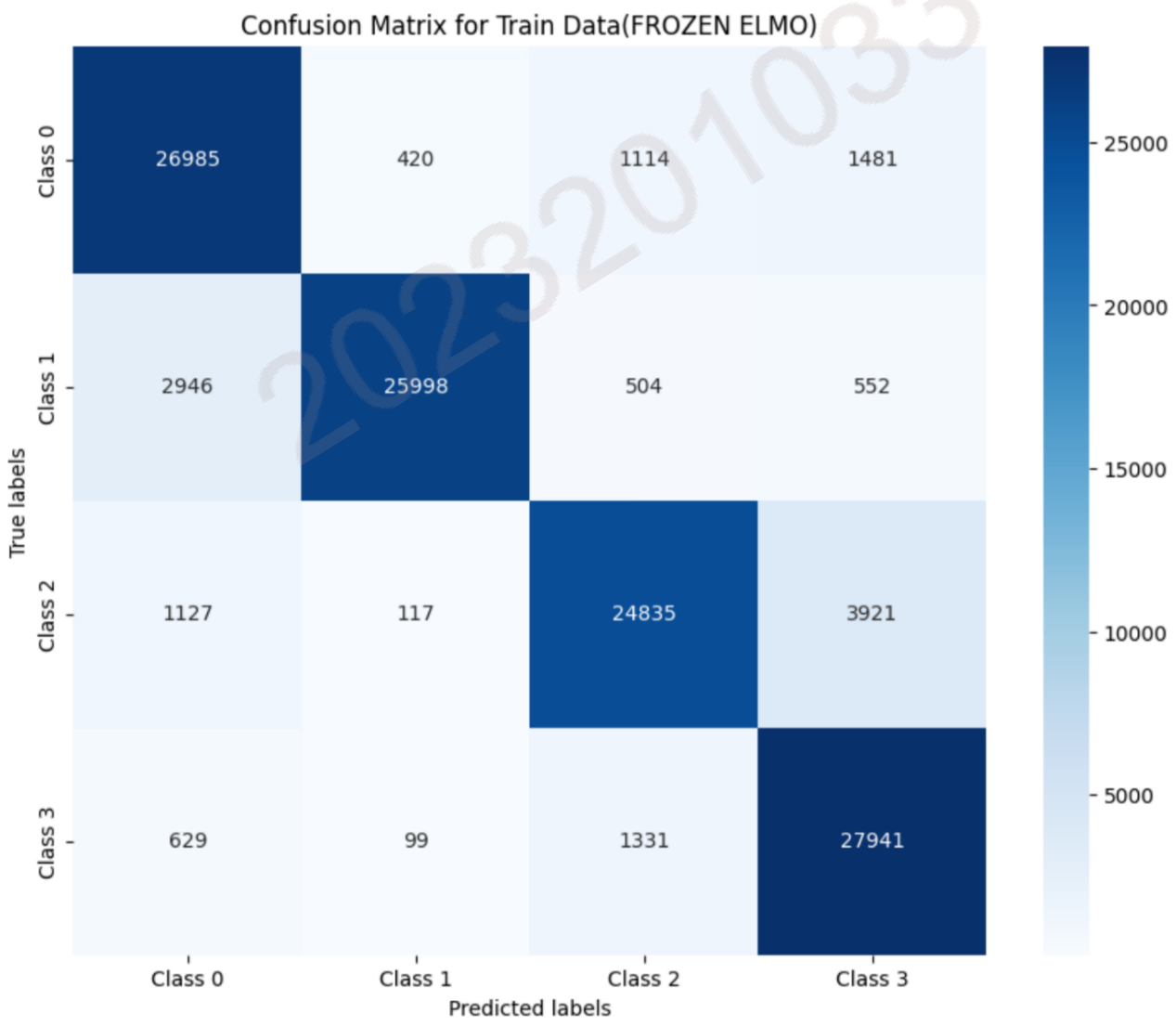
TRAIN - DATA CLASSIFICATION REPORT

	precision	recall	f1-score	support
0	0.85	0.90	0.87	30000
1	0.98	0.87	0.92	30000
2	0.89	0.83	0.86	30000
3	0.82	0.93	0.87	30000
accuracy			0.88	120000
macro avg	0.89	0.88	0.88	120000
weighted avg	0.89	0.88	0.88	120000

- Class 1 has the highest precision, suggesting that predictions for this class are highly reliable, while Class 3 has the highest recall, states that it captures most of the actual Class 3 instances.
- Class 3 has the lowest precision, shows that more false positives for this class
- The overall accuracy and the macro and weighted averages across precision, recall, and F1-score are all 0.88, which is a strong performance but slightly lower than what was achieved on the test set when the lambda parameters were trainable.

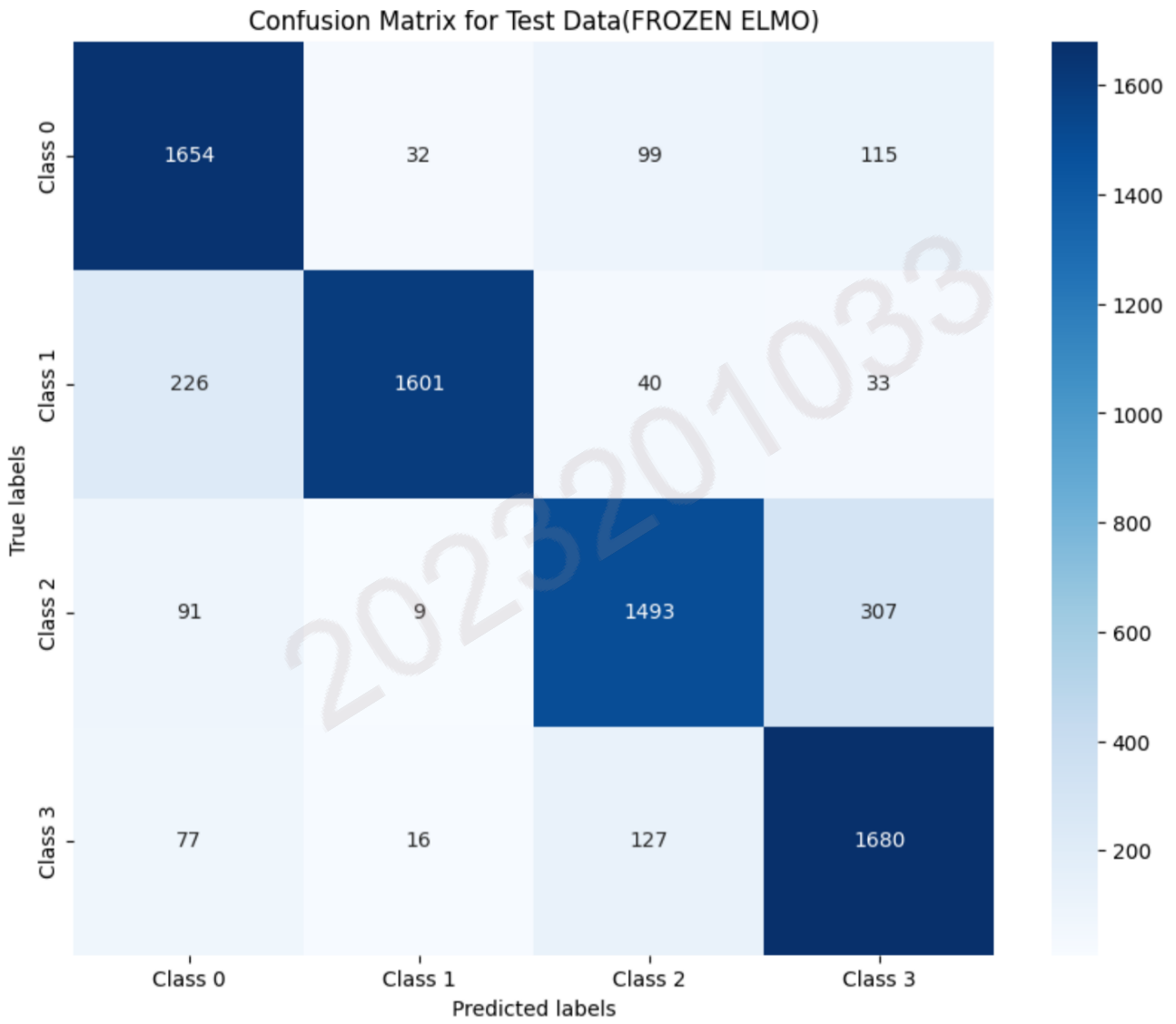
The slightly lower training accuracy compared to the test accuracy with trainable lambdas is due to the model's reduced capacity to fine-tune the internal feature combination without the trainable lambda parameters, which is preventing it from capturing the full complexity of the training data.

CONFUSION MATRIX TRAINING DATA



- Class 0 and Class 3 have the highest number of correct predictions with 26985 and 27,941, respectively, indicative of a strong true positive rate. Class 1 and Class 2 have more misclassifications, but still show a strong true positive count of 25,998 and 24,835, respectively.
- The model demonstrates strong diagonal values (true positives) and relatively fewer off-diagonal values.
- However, numbers are less compared to those of trainable lambdas.

CONFUSION MATRIX TEST_DATA



- Class 0 and Class 3 have the highest number of correct predictions with 1654 and 1680, respectively, indicative of a strong true positive rate. Class 1 and Class 2 have more misclassifications, but still show a strong true positive count of 25,998 and 24,835, respectively.
- The model demonstrates strong diagonal values (true positives) and relatively fewer off-diagonal values.
- However, numbers are less compared to those of trainable lambdas.
- If we closely observe, each of them has confused with adjacent classes.

ELMO - LEARNABLE FUNCTION

Parameter setting

Here, I have removed lambda and instead added a non-linear function. Now, this function will learn the context in this setting.

```
self.learnable_function = nn.Sequential(
    nn.Linear(hidden_dim * 3, hidden_dim),
    nn.ReLU(inplace=True)
)
```

TEST - DATA CLASSIFICATION REPORT

	precision	recall	f1-score	support
0	0.92	0.90	0.91	1900
1	0.96	0.97	0.96	1900
2	0.88	0.83	0.86	1900
3	0.85	0.89	0.87	1900
accuracy			0.90	7600
macro avg	0.90	0.90	0.90	7600
weighted avg	0.90	0.90	0.90	7600

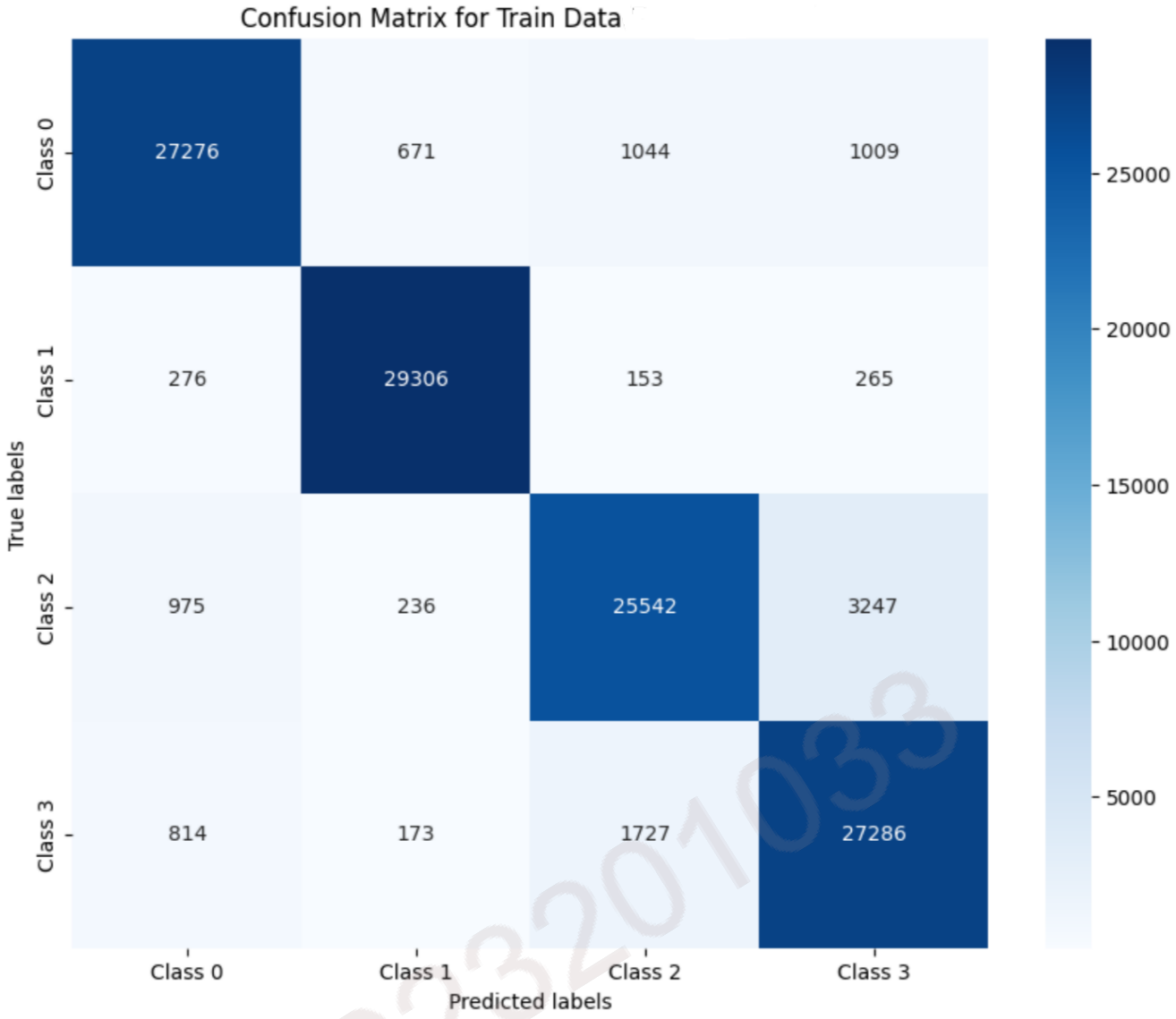
- The precision ranges from 0.85 to 0.96, and recall from 0.83 to 0.97, shows that the model is consistently predicting most classes well.
- The overall accuracy, macro average, and weighted average are all at 0.90, indicates a well-balanced and high-performing model across all classes.
- This has allowed the model to effectively learn and integrate context into its predictions.

TRAIN - DATA CLASSIFICATION REPORT

	precision	recall	f1-score	support
0	0.93	0.91	0.92	30000
1	0.96	0.98	0.97	30000
2	0.90	0.85	0.87	30000
3	0.86	0.91	0.88	30000
accuracy			0.91	120000
macro avg	0.91	0.91	0.91	120000
weighted avg	0.91	0.91	0.91	120000

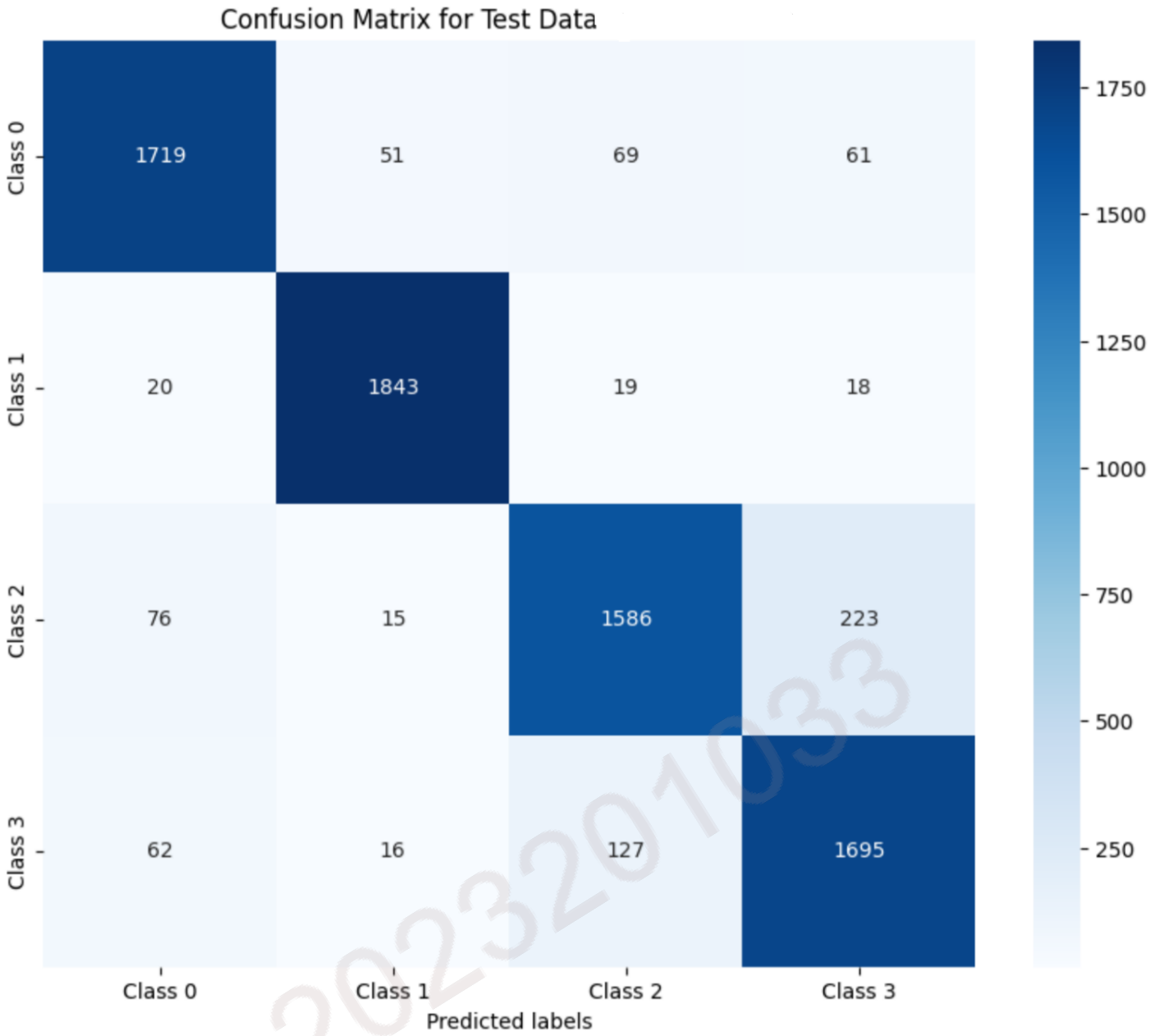
- Class 1 shows the highest precision and F1-score, shows us that it is the easiest for the model to predict correctly.
- Class 2 and Class 3 show slightly lower precision and F1-scores, indicating a bit more difficulty in correctly predicting those classes.
- The overall accuracy, as well as macro and weighted averages, are all 0.91, which is a strong performance but not perfect, p**ossibly due to the model's generalization efforts to prevent overfitting, or because the non-linear function used in place of lambda is less optimal for this training data.**

CONFUSION MATRIX TRAINING DATA



- Class 1 and Class 2 have the highest number of correct predictions, shows that the model is particularly adept at identifying these classes.
- The distribution of predictions, especially the correct ones along the diagonal, confirms the model's discrimination ability.

CONFUSION MATRIX TEST DATA



- Model has a strong diagonal indicating correct classifications, with Class 1 performing exceptionally well with 1843 correct predictions.
- Some confusion between Class 2 and Class 3, as seen by the 223 instances where Class 2 was mistaken for Class 3.
- Misclassifications are relatively low for Class 0 and Class 1, demonstrating the model's accuracy for these classes.

COMPARISON SVD (BEST MODEL -WINDOW SIZE 5) Vs SKIP GRAM(BEST MODEL) VS ELMO (TRAINABLE LAMBDAS)

SVD

ACCURACY, RECALL AND F1 VALUES WHEN WINDOW SIZE =5

```
Test Accuracy: 86.84%
Training Set Performance:
Accuracy: 0.90
F1 Score: 0.90
Precision: 0.90
Recall: 0.90

Test Set Performance:
Accuracy: 0.87
F1 Score: 0.87
Precision: 0.87
Recall: 0.87
(myenv) PS C:\Users\nithi\OneDrive\Desktop\IIIT HYDERA
```

SKIPGRAM

```
Test Accuracy: 89.55%
Training Set Performance:
Accuracy: 0.91
F1 Score: 0.91
Precision: 0.91
Recall: 0.91
Confusion Matrix:

Test Set Performance:
Accuracy: 0.90
F1 Score: 0.90
Precision: 0.90
Recall: 0.90
Confusion Matrix:
(myenv) PS C:\Users\nithi\OneDrive\Desktop\IIIT HYDERABAD\sem2\NLP\20232010
```

ELMO

	precision	recall	f1-score	support
0	0.92	0.91	0.92	1900
1	0.96	0.97	0.97	1900
2	0.89	0.85	0.87	1900
3	0.86	0.90	0.88	1900
accuracy			0.91	7600
macro avg	0.91	0.91	0.91	7600
weighted avg	0.91	0.91	0.91	7600

Out of all the 3, Elmo seems to be the clear winner.

why?

1. **Accuracy Metrics**: Elmo often achieves higher accuracy metrics compared to SVD or Skip-Gram. For instance, if Elmo shows an accuracy of 91% on a test set, SVD and Skip-Gram might display lower figures, such as 86% or 89%, respectively.
2. **F1-Score**: Elmo could present F1-scores around 0.91 or higher, while SVD and Skip-Gram is lagging with scores 87% and 90% suggesting Elmo's superior balance of precision and recall.
3. **Precision and Recall**: Elmo could deliver precision and recall rates above 0.91, which may be noticeably higher than those achieved with SVD or Skip-Gram models (best models), which might hover around 0.87-0.89.
4. **Support**: All models might have been trained and tested on datasets with equal class distribution (support), but Elmo's ability to maintain high performance across all classes often stands out more distinctly.

Seeing the data, we know ELMO is the winner, but then why not a considerable difference?

Training and Task Specificity on the Same Dataset

- Elmo is both pre-trained and fine-tuned on the same dataset, the distinction between Elmo and simpler models like Skip-Gram may not be significant.
- Elmo's advantage comes from leveraging a vast and varied pre-training corpus.
- My custom Elmo still manages to achieve superior or comparable results due to its context-aware architecture.
- When Elmo is used in the traditional way—pre-trained on a large, diverse corpus and then fine-tuned on a specific task's dataset—the benefits of its deep, contextualized representations become much more apparent.

Which Hyper parameter setting gave best result and why?

- In my case specifically, both are giving a same accuracy of 91 percent in test set and a training accuracy of 94 percent in training set(trainable lambda) and 92 percent (learning function respectively).
- The lambda layer is allowing slightly more capacity for the model to learn from the training data compared to the learnable function (because training set accuracy is more)
- we cannot confirm this, because then we might have to change the lr rate of learnable function and try again.
- **So, overall, in my case if you consider training accuracy also then trainable lambda performed better.**

What should have happened ideally?

A learnable function is expected to perform better than trainable lambdas in Elmo's architecture because,

1. A learnable function, particularly if it includes non-linear activation functions, can capture more complex patterns and interactions between the different levels of representation in Elmo.
2. Trainable lambdas are just scaling factors, which, while useful, offer less flexibility and adaptability.
3. It can theoretically learn to emphasize or de-emphasize certain features as needed, something that a simple set of trainable lambdas may not do as effectively.

May be because the function I chose might not be a better replacement for trainable lambda.

Best hyper Parameter settings

1. Trainable lambda
2. Lr rate, 0.001
3. epochs : 5

Dimension

```
Using device: cuda
```

```
ELMo(
  (embedding): Embedding(78717, 100)
  (dropout): Dropout(p=0.5, inplace=False)
  (embedding_projection): Linear(in_features=100, out_features=256, bias=True)
  (forward_lstm1): LSTM(256, 256, batch_first=True)
  (forward_lstm2): LSTM(256, 256, batch_first=True)
  (backward_lstm1): LSTM(256, 256, batch_first=True)
  (backward_lstm2): LSTM(256, 256, batch_first=True)
  (output_forward): Linear(in_features=256, out_features=78717, bias=True)
  (output_backward): Linear(in_features=256, out_features=78717, bias=True)
  (combined_output_projection): Linear(in_features=512, out_features=256, bias=True)
)
```

**********END OF REPORT**********