# Improving Peer-Review Score Prediction with Semi-Supervised Learning and Denoising Networks

Panitan Muangkammuen[1], Fumiyo Fukumoto[2], Jiyi Li[2], and Yoshimi Suzuki[2]
[1]Integrated Graduate School of Medicine, Engineering, and Agricultural Sciences
[2]Interdisciplinary Graduate School
University of Yamanashi
{g21dts04,fukumoto,jyli,ysuzuki}@yamanashi.ac.jp

## Abstract

Peer review aspect score prediction (PASP) is a valuable tool for improving the efficiency and effectiveness of academic peer review processes. However, the limited availability of labeled peer review data can pose a challenge for traditional supervised learning approaches. In this paper, we present a novel semi-supervised learning (SSL) method for PASP that leverages contextual features from unlabeled data to improve performance. Our approach incorporates the Long-Short Transformer (Transformer-LS), a transformer for long sequences with linear complexity, into the Γ-model, a variant of the Ladder network that utilizes a denoising autoencoder to reconstruct the input data from a corrupted version. By minimizing the reconstruction error of the auxiliary unlabeled data, it can help in training the classifier. We empirically demonstrate the superior performance of our system compared to the supervised and naive semi-supervised learning baselines on the PeerRead benchmark.

## 1 Introduction

The increasing number of submissions to AI-related international conferences and journals has made the review process more challenging. Automatic peer-review aspect score prediction (PASP) is a valuable tool for improving the efficiency and effectiveness of the review process by providing reviewers and authors with a numeric score for different qualities of a paper, such as clarity and originality. The PeerRead dataset [1] is the first publicly available collection of scientific peer reviews for research purposes and has been used in a variety of applications, including paper acceptance classification [2, 3, 4], review aspect score prediction [5, 6], citation recommendation [7], and citation count prediction [8].

Previous work on PASP has heavily relied on supervised learning techniques [1, 5]. However, the available annotated datasets for this task are very restricted, which limits the overall performance of PASP models. To address this issue and improve PASP performance, we propose a semi-supervised learning (SSL) method that leverages contextual features from a larger, unlabeled dataset. Semi-supervised learning has been widely used in various natural language processing (NLP) tasks, including classification [9, 10], sequence labeling [11, 12], and parsing [13, 14]. It has been shown to be effective in model learning by leveraging a large amount of unlabeled data to compensate for the lack of labeled data. Semi-supervised learning is particularly useful for PASP, as a vast number of scholarly papers are available online and can be easily obtained as unlabeled data.

Recently, transformers [15] have achieved state-of-the-art results in a wide range of NLP tasks. However, transformer-based models are unable to process long sequences, such as academic papers, due to their self-attention operation, which scales quadratically with the sequence length. In this paper, we propose a semi-supervised learning technique for PASP that is capable of handling long sequences. Our approach is based on the combination of ladder networks (LNs) [16, 17] and the Long-short transformer (Transformer-LS) [18]. Ladder networks are a type of deep denoising autoencoder that incorporates skip connections and reconstruction targets at intermediate layers, while Transformer-LS is a transformer with a self-attention mechanism that is efficient for modeling long sequences with linear complexity. We propose the Γ-

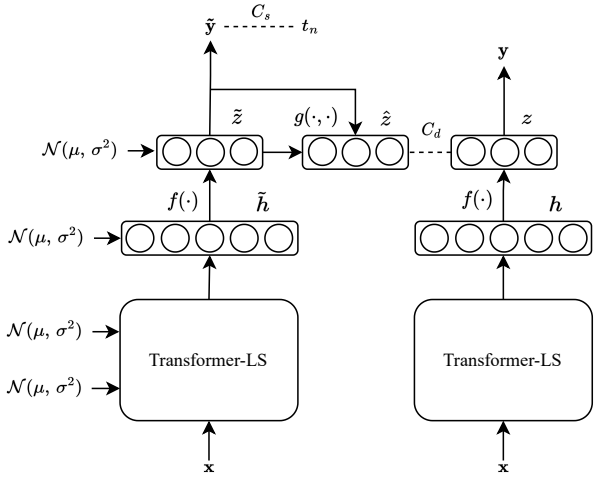


**Figure 1** Γ-TLS architecture. The corrupted path shown on the left-hand side shares the Transformer-LS's weights and mapping $f$ with the clean path on the right-hand side.

Transformer-LS (Γ-TLS), which integrates a Transformer-LS into the Γ-model [16], a variant of ladder networks. The unsupervised component of Γ-TLS utilizes a denoising autoencoder to help focus on relevant features derived from supervised learning.

To the best of our knowledge, our work is one of the first applications of SSL to the PASP task. Specifically, our contributions are as follows:

1. We propose Γ-TLS for PASP that incorporates a Transformer-LS into SSL by training the model using labeled and unlabeled data simultaneously.
2. The experimental results show that Γ-TLS outperforms the supervised learning baselines and naive SSL methods on the PeerRead benchmark.

## 2 Γ-Transformer-LS (Γ-TLS)

To overcome the limitation of the vanilla transformer [15] for long sequences, we adopt the Transformer-LS as the encoder of our framework. Transformer-LS is more memory and computationally efficient than the previous larger models, Longformer [19] and Transformer-XL [20]. For the SSL technique, we choose a denoising network called the Γ-model [16], which is a variant of ladder networks (LNs). The Γ-model eliminates most of the decoder, retaining only the top layer, which allows it to be easily integrated into any network without implementing a separate decoder. The encoder in the Γ-model still includes both the clean and corrupted paths, as in the full ladder network (LN).

| Aspect | #Neg / #Pos | Total |
|---|---|---|
| Clarity (Clr) | 39 / 97 | 136 |
| Originality (Ori) | 58 / 78 | 136 |
| Impact (Imp) | 110 / 22 | 132 |
| Meaningful comparison (Com) | 80 / 52 | 132 |
| Soundness correctness (Cor) | 54 / 82 | 136 |
| Substance (Sub) | 66 / 70 | 136 |
| Overall recommendation (Ova) | 76 / 60 | 136 |

**Table 1** Statistics of the ACL Dataset.

Figure 1 illustrates the Γ-Transformer-LS (Γ-TLS). Let $\mathbf{x}$ be the input and $y$ be the output with targets $t$. The supervised data of size $N$ consists of pairs $\{\mathbf{x}(n), t(n)\}$, where $1 \leq n \leq N$. The unsupervised data of size $M$ has only input $\mathbf{x}$ without the targets $t$, an $\mathbf{x}(n)$, where $N+1 \leq n \leq N+M$. The network comprises two forward passes, the clean path, and the corrupted path. The clean path, illustrated on the right-hand side in Figure 1, produces clean representation $\mathbf{z}$ and clean output $\mathbf{y}$, given by:

$$\begin{aligned} \mathbf{z} &= f(\mathbf{h}) = N_B(\mathbf{W}\mathbf{h}) \\ \mathbf{y} &= \phi(\gamma(\mathbf{z}+\beta)) \\ \mathbf{h} &= TLS(\mathbf{x}), \end{aligned} \tag{1}$$

where $\mathbf{h}$ denotes the hidden representation obtained from Transformer-LS ($TLS$), $\mathbf{W}$ is the weight matrix of the linear transformation $f$, and $N_B$ indicates a batch normalization. $\phi$ refers to an activation function, where $\beta$ and $\gamma$ are trainable scaling and bias parameters, respectively.

The corrupted representation $\tilde{\mathbf{z}}$ and corrupted output $\tilde{\mathbf{y}}$ are produced by adding Gaussian noise $\mathbf{n}$ in the corrupted path (left-hand side of Figure 1). The noise $\mathbf{n}$ is applied to the output of each layer of the Transformer-LS ($TLS$):

$$\begin{aligned} \tilde{\mathbf{z}} &= f(\tilde{\mathbf{h}}) + \mathbf{n} \\ \tilde{\mathbf{y}} &= \phi(\gamma(\tilde{\mathbf{z}}+\beta)) \\ \tilde{\mathbf{h}} &= TLS(x) + \mathbf{n}. \end{aligned} \tag{2}$$

The supervised cost $C_s$ is the average negative log-probability of the corrupted output $\tilde{\mathbf{y}}$ matching the target $t_n$ given the input $\mathbf{x}_n$:

$$C_s = -\frac{1}{N}\sum_{n=1}^{N} \log P(\tilde{\mathbf{y}} = t_n | \mathbf{x}_n), \tag{3}$$

Given the corrupted $\tilde{\mathbf{z}}$ and prior information $\tilde{\mathbf{y}}$, the denoising function $g$ reconstructs the denoised $\hat{\mathbf{z}}$:

| Metric | Models | Clr | Ori | Imp | Com | Cor | Sub | Ova | Avg. |
|---|---|---|---|---|---|---|---|---|---|
| Acc. | CNN | 0.721 | 0.595 | 0.833 | 0.636 | 0.640 | 0.566 | 0.611 | 0.657 |
| | VAT | 0.728 | 0.669 | 0.841 | 0.614 | 0.669 | 0.662 | 0.654 | 0.691 |
| | HAN | 0.720 | **0.690** | 0.841 | 0.674 | 0.684 | 0.654 | 0.692 | 0.708 |
| | Multi-task | 0.736 | 0.661 | **0.864** | **0.713** | **0.698** | 0.617 | 0.670 | 0.708 |
| | Transformer-LS | 0.735 | 0.646 | 0.856 | 0.696 | 0.647 | 0.677 | 0.684 | 0.706 |
| | Γ-TLS (Ours) | **0.757** | 0.654 | 0.856 | 0.703 | **0.698** | **0.706** | **0.728** | **0.729** |
| F1. | CNN | 0.482 | 0.442 | 0.455 | 0.497 | 0.513 | 0.503 | 0.463 | 0.479 |
| | VAT | 0.489 | **0.620** | 0.536 | 0.398 | 0.620 | 0.660 | 0.603 | 0.561 |
| | HAN | 0.493 | 0.613 | 0.490 | 0.608 | 0.661 | 0.578 | 0.664 | 0.587 |
| | Multi-task | **0.581** | 0.461 | **0.621** | **0.671** | **0.673** | 0.612 | 0.633 | 0.607 |
| | Transformer-LS | 0.508 | 0.549 | 0.583 | 0.628 | 0.557 | 0.594 | 0.662 | 0.583 |
| | Γ-TLS (Ours) | 0.553 | 0.558 | 0.567 | 0.661 | 0.639 | **0.625** | **0.717** | **0.617** |

**Table 2** Experimental results. The best result is in bold, and the 2nd best is underlined.

$$\begin{aligned} \hat{\mathbf{z}} &= g(\tilde{\mathbf{z}}, \mathbf{u}) \\ \mathbf{u} &= N_B(\tilde{\mathbf{y}}), \end{aligned} \tag{4}$$

where $g$ is identical to the one of the LN [16] consisting of its own learnable parameters. The unsupervised denoising cost function is given by:

$$C_d = \frac{1}{N+M} \sum_{n=1}^{N+M} \frac{\lambda}{d} \|\mathbf{z}_n - N_B(\hat{\mathbf{z}}_n)\|, \tag{5}$$

where $\lambda$ is a coefficient for unsupervised cost, and $d$ refers to the width of the output layer. The final cost $C$ is given by:

$$C = C_s + C_d$$

# 3 Experiments

## 3.1 Setup

**Data** The ACL 2017 dataset, included in PeerRead [1], is used as evaluation data for our PASP system. The ACL dataset consists of 7 different aspects of scores as listed in Table 1. These aspect scores were derived from a mean of multiple reviews and classified into two categories: positive (scores of 4 or higher) and negative (scores lower than 4). Although the PeerRead dataset contains both paper and review texts, we only used the papers to predict the aspect scores. We utilized the first 8,192 tokens of the paper as the input. We used SciVocab [21] WordPiece vocabulary for tokenization. For the unlabeled data, we used the ACL papers from ScisummNet Corpus [1)] [22], which provides 999 papers in the ACL anthology.

To evaluate all systems, we employed a 5-fold cross-validation strategy, in which the final result was calculated as the average of the five folds. As the evaluation metrics, we utilized both accuracy and Macro F1 score. This allows us to comprehensively assess the performance of our systems in terms of both the proportion of correct predictions and the balance between precision and recall.

**Baseline models** The competitor algorithms that are used as baselines for our model are the following:

- **CNN** - We implemented a CNN model similar to one in PeerRead [1]. The outputs from the CNN model are passed through a max pooling layer and finally through the final linear layer.
- **VAT** [9] - An SSL method that exploits information from unlabeled data by applying perturbations to the word embeddings in a neural network. The model utilizes LSTM to learn from sequential inputs.
- **HAN** [23] - A hierarchical attention network for document classification. The model consists of two levels of attention mechanisms at the word and sentence levels to construct the document representation.
- **Multi-task** [5] - A multi-task approach that automatically selects shared network structures and other review aspects as auxiliary resources. The model is based on the CNN text classification model.
- **Transformer-LS** [18] - A transformer for modeling long sequences with linear complexity. We used the

1) https://cs.stanford.edu/~myasu/projects/scisumm_net/

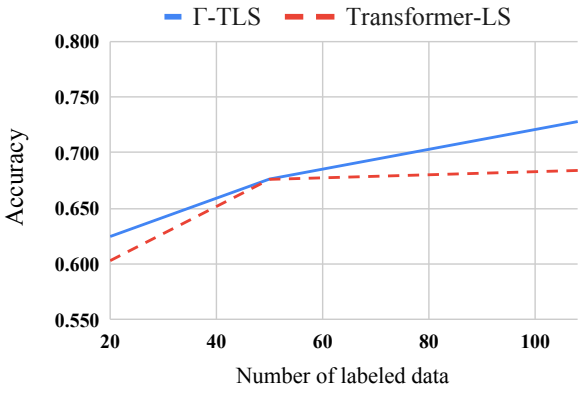


**Figure 2** Accuracy against the number of labeled data on *Overall recommendation* score prediction. The number of unlabeled data is fixed to 999 for Γ-TLS.

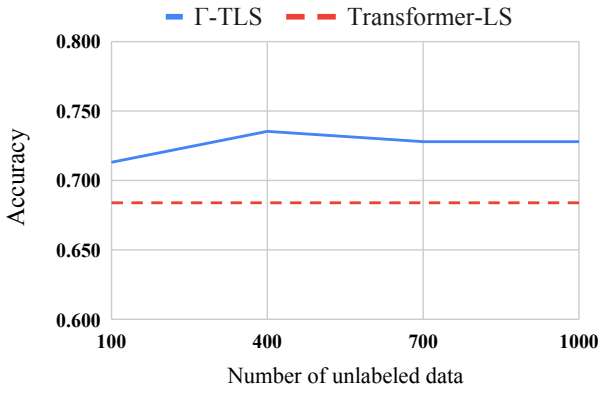


**Figure 3** Accuracy against the number of unlabeled data on *Overall recommendation* score prediction.

output of the last layer of the [CLS] token as the document representation for the classifier.

The implementation details are shown in Appendix A Implementation details.

### 3.2 Results

The results are listed in Table 2. Our model, Γ-TLS, demonstrated superior performance in several aspects compared to the baseline models. When evaluated using the accuracy metric, Γ-TLS outperformed the baseline models on four aspects: *Clarity*, *Soundness Correctness*, *Substance*, and *Overall Recommendation*. Additionally, Γ-TLS outperformed the baseline models when evaluated using the Macro F1 score metric on two aspects: *Subtance* and *Overall Recommendation*. Overall, Γ-TLS performed the best out of all the models across an average of seven aspects on both metrics.

Additionally, we observe that the Transformer-LS outperforms the CNN by almost 5% in accuracy and 10% in Macro F1 score, which shows that the attention mechanism is relatively more effective for modeling the documents. By applying a hierarchical structure, SSL, or multi-task learning technique, the performance is also further improved.

### 3.3 Ablation study

In comparison to the Transformer-LS model, the incorporating of a denoising network (ladder network) into Transformer-LS resulted in improved performance in almost every aspect, except for *Impact* on the accuracy and *Impact* and *Meaningful Comparison* on Macro F1 score. On average, Γ-TLS outperformed Transformer-LS by 2.3% in accuracy and 3.4% in terms of Macro F1 score metric. This indicates that our assumption, leveraging contextual features from unlabeled data, helps to improve performance.

We also investigated how the number of labeled data used for training affects the overall performance. As shown in Figure 2, increasing the number of labeled data tends to improve the performance of both Γ-TLS and Transformer-LS, with the exception of a labeled data count of 50, where the results were not significantly different. Overall, Γ-TLS consistently outperformed Transformer-LS, which shows that our proposed SSL method is stably effective on small training data. In addition, the effect of the number of unlabeled data on model performance was examined, as shown in Figure 3. The results indicate that Γ-TLS's performance improved when the number of unlabeled data was increased from 100 to 400, but saw no further improvement beyond that point. Our model, Γ-TLS, still outperforms the Transformer-LS by using only 100 unlabeled data.

## 4 Conclusion

In this paper, we focused on the task of automated peer review aspect score prediction (PASP) and proposed a novel method called Γ-TLS. The method integrates the Transformer-LS model with the denoising network, the Γ-model of ladder networks. Our experimental results showed that Γ-TLS outperformed the baseline models on average accuracy and F1 score. In future research, we plan to investigate ways to leverage related information between aspects for our model, as well as to generate more knowledgeable and explainable review comments.

## Acknowledgement

This work was supported by JKA and JST SPRING, Grant Number JPMJSP2133.

## References

[1] Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. A dataset of peer reviews (PeerRead): Collection, insights and NLP applications. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)**, pp. 1647–1661, 2018.

[2] Tirthankar Ghosal, Rajeev Verma, Asif Ekbal, and Pushpak Bhattacharyya. DeepSentiPeer: Harnessing sentiment in review texts to recommend peer review decisions. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 1120–1130, 2019.

[3] Gideon Maillette de Buy Wenniger, Thomas van Dongen, Eleri Aedmaa, Herbert Teun Kruitbosch, Edwin A. Valentijn, and Lambert Schomaker. Structure-tags improve text classification for scholarly document quality prediction. In **Proceedings of the First Workshop on Scholarly Document Processing**, pp. 158–167, 2020.

[4] Panagiotis Fytas, Georgios Rizos, and Lucia Specia. What makes a scientific paper be accepted for publication? In **Proceedings of the First Workshop on Causal Inference and NLP**, pp. 44–60, 2021.

[5] Jiyi Li, Ayaka Sato, Kazuya Shimura, and Fumiyo Fukumoto. Multi-task peer-review score prediction. In **Proceedings of the First Workshop on Scholarly Document Processing**, pp. 121–126, 2020.

[6] Qingyun Wang, Qi Zeng, Lifu Huang, Kevin Knight, Heng Ji, and Nazneen Fatema Rajani. ReviewRobot: Explainable paper review generation based on knowledge synthesis. In **Proceedings of the 13th International Conference on Natural Language Generation**, pp. 384–397, 2020.

[7] Chanwoo Jeong, Sion Jang, Hyuna Shin, Eunjeong L. Park, and Sungchul Choi. A context-aware citation recommendation model with bert and graph convolutional networks, 2019.

[8] Thomas van Dongen, Gideon Maillette de Buy Wenniger, and Lambert Schomaker. SChuBERT: Scholarly document chunks with BERT-encoding boost citation count prediction. In **Proceedings of the First Workshop on Scholarly Document Processing**, pp. 148–157, 2020.

[9] Takeru Miyato, Andrew M. Dai, and Ian Goodfellow. Adversarial training methods for semi-supervised text classification, 2016.

[10] Changchun Li, Ximing Li, and Jihong Ouyang. Semi-supervised text classification with balanced deep representation distributions. In **Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**, pp. 5044–5053, 2021.

[11] Michihiro Yasunaga, Jungo Kasai, and Dragomir Radev. Robust multilingual part-of-speech tagging via adversarial training. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)**, pp. 976–986, 2018.

[12] Luoxin Chen, Weitong Ruan, Xinyue Liu, and Jianhua Lu. SeqVAT: Virtual adversarial training for semi-supervised sequence labeling. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 8801–8811, 2020.

[13] Xiao Zhang and Dan Goldwasser. Semi-supervised parsing with a variational autoencoding parser. In **Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies**, pp. 40–47, 2020.

[14] Kyungtae Lim, Jay Yoon Lee, Jaime G. Carbonell, and Thierry Poibeau. Semi-supervised learning on meta structure: Multi-task tagging and parsing in low-resource scenarios. In **The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI**, pp. 8344–8351. AAAI Press, 2020.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[16] Antti Rasmus, Harri Valpola, Mikko Honkala, Mathias Berglund, and Tapani Raiko. Semi-supervised learning with ladder networks, 2015.

[17] Harri Valpola. From neural pca to deep unsupervised learning, 2014.

[18] Chen Zhu, Wei Ping, Chaowei Xiao, Mohammad Shoeybi, Tom Goldstein, Anima Anandkumar, and Bryan Catanzaro. Long-short transformer: Efficient transformers for language and vision, 2021.

[19] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020.

[20] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 2978–2988, 2019.

[21] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text. In **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)**, pp. 3615–3620, 2019.

[22] Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R. Fabbri, Irene Li, Dan Friedman, and Dragomir R. Radev. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks, 2019.

[23] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In **Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 1480–1489, 2016.

# A Implementation details

## A.1 CNN

We used a simple MLP with a single hidden layer of 128 neurons with the max pooling of a single 1D-CNN layer of 128 filters and window width 5. We used a random initialization for the word embeddings size of 128 and trained it with the model. We trained the model using AdamW optimizer on a linear scheduler, a learning rate of 1e-4 with a batch size of 8.

## A.2 HAN

We set the max sentence length to 100 tokens and the max number of sentences to 600. We used a bidirectional single-layer GRU size of 100 with an attention mechanism to aggregate the representation on both word and sentence levels. we also used a random initialization for the word embeddings size of 300. The model was trained on AdamW optimizer, learning rate of 5e-5, and batch size of 8.

## A.3 VAT

### A.3.1 Recurrent LM Pre-training

We used a unidirectional single-layer LSTM with 128 hidden units. The dimension of word embedding was 128. For the optimization, we used the Adam optimizer with a batch size of 32, an initial learning rate of 0.001, and a 0.9999 learning rate decay factor. We trained for 50 epochs. We applied gradient clipping with norm set to 5.0. We used dropout on the word embedding layer and an output layer with a 0.5 dropout rate.

### A.3.2 Model Training

We added a hidden layer between the softmax layer for the target and the final output of the LSTM. The dimension is set to 30. For optimization, we also used the Adam optimizer, with a 0.001 initial learning rate and 0.9998 exponential decay. Batch sizes are set to 32 and 96 for calculating the loss of virtual adversarial training. We trained for 30 epochs. applied gradient clipping with the norm as 5.0.

## A.4 Multi-task

We modified the model from performing a regression task to a classification task by changing the output layer. We used CNN with 64 filters and filter width of 2. We used fastText as initial word embeddings. The hidden dimension was 1024. We trained the model using Adam optimizer with learning rate 0.001 and batch size of 8. We trained all of the candidate multi-task models for two auxiliary tasks to find the best one.

## A.5 Transformer-LS

We used two layers of transformer-ls size 256 with 4 attention heads. The local window attention was set to 128. A [CLS] token was used as a global token. We used dropout and attention dropout of 0.1. We trained the model using AdamW optimizer on a linear scheduler with batch size 8. We tuned the learning rate in the range of {1e-2, 1e-3, 1e-4}

## A.6 Γ-TLS

We used the same architecture as the Transformer-LS (A.5). The denoising cost multipliers $\lambda$ is set to 1. We tuned the std of the Gaussian corruption noise in the range of {0.1, 0.2, 0.3}. We also tuned the learning rate in the range of {1e-2, 1e-3, 1e-4}. Batch size is set to 8 for both labeled and unlabeled data, 16 in total.